Just For Your TRS-80® Model 100 and Tandy 2000

PCM™

$3
$3.95 Canadian
Vol. II No. 2
August 1984

The Magazine for Professional Computing Management

Coding Sensitive Files

Interfacing PoCo to CoCo

Touring Dow Jones

Plus
A Bar Chart Program for your 2000

and Bar Code Listings for your 100

ATTENTION TRS-80 MODEL 100 OWNERS

AT LAST! SOMETHING FOR NOTHING

(That means absolutely free!)

WHAT?

Just call us on the toll free number listed here and we'll send you two free programs on a cassette for your TRS 80 Model 100. They are:
File Manager: lets you kill, rename, and make files invisible at the touch of a function key plus 6 other features.
Restore: lets you restore lost text files from cold start or crashed computer.

Nifty? You bet! But you're probably wondering why we're doing this? For a very good reason. Keep reading.

WHY?

Because we want to put in your hands information on a very important new product that you've been waiting for, ROM Programs! Ever since you bought your TRS 80 Model 100 you've heard about programs in ROM. Well, they're here, ready to snap in and go! Giving you the power you've been waiting for. We believe you'll be so excited about our ROM programs that we're willing to give you FREE software just to tell you about them. So call today to order your free programs. You'll also receive all the details about this revolution in Model 100 software.

1-800-641-4645 Ext. 207

PORTABLE COMPUTER SUPPORT GROUP

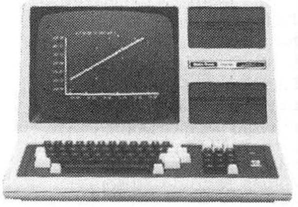
11035 Harry Hines, Suite 207
Dallas, Texas 75229 (214) 351-0564

From Computer Plus to YOU . . .

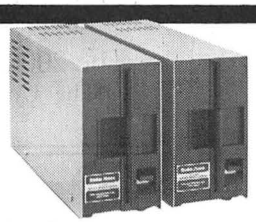
PLUS *after* PLUS *after* PLUS

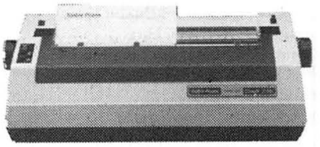
Model 100 8K $495
Model 100 24K $625

Color Computer II
w/64K Ext. Basic $205


Model 4 16K $679
Model 4 64K
2 Disk & RS232 $1020

DMP120 $385
DMP200 $520


Color Computer Disk Drive
Drive 0 $289 Drive 1 $220


DWP210 $489

BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS

COMPUTERS	
Model 4 Portable	
64K 2 Drives	1020
Pocket Computer 3	89
Pocket Computer 4	59
Model 2000 2 Drive	2299
Model 2000 W/Hard Disk	3599
Model 12 64K 1 Drive	2360
Model 12 64K 2 Drive	2950
Model 16B 1Dr 256K	3375
Model 16B 2Dr 256K	3965
MODEMS	
Hayes Smartmodem II	215
Hayes Smartmodem 1200	505
Novation Smartcat 1200	410
Novation J-Cat	115
AC-3	125
DC Modem I	89
DC Modem II	160
DC Modem 2212	315

PRINTERS	
Silver Reed EXP400 D.W.	309
Silver Reed EXP550 D.W.	430
DWP410	1159
Gemini Powertype	345
CGP115	159
CGP220 Ink Jet	545
DMP110	305
DMP420	735
Toshiba 1340 (24 wire head)	779
Gemini 10X	289
Gemini 15X	409
Panasonic P1091	315
Smith Corona Fastext	190
Prowriter 8510	345
Okidata	CALL
Epson	CALL
DISK DRIVES	
Model III/4 Drive 0	450
Model III/4 Drive 1	195
Model III/4 Drive 2	259
Model III/4 Drive 3	220
Hard Disk M2000	1525
ETC.	
CCR-81 Recorder	52
Cassette Tapes (10 Pk)	9.95
16K Ram Chips	25
64K Ram Chips	62.95
Model 4 64K Upgrade Kit	79.95
M100 Bar Code Reader	89
64K Printer Buffer	199
Printer Stand	25
Printer Cables	
Printer Ribbons	
Printer Paper	
Computer Books	

Radio Shack software 10% off.
Send for complete listing of
brand name software and hardware.

CALL TOLL FREE
1-800-343-8124

- *LOWEST POSSIBLE PRICES*
- *BEST POSSIBLE WARRANTY*
- *KNOWLEDGEABLE SALES STAFF*
- *TIMELY DELIVERY*
- *SHOPPING CONVENIENCE*

VISA AMERICAN EXPRESS MasterCard

computer plus

P.O. Box 1094
480 King Street
Littleton, MA 01460

SINCE 1973

IN MASSACHUSETTS CALL (617) 486-3193

TRS-80 is a registered trademark of Tandy Corp.

PCM

The Magazine for Professional Computing Management

Vol. 2, Issue 2 August 1984

Editor and Publisher
Lawrence C. Falk

Managing Editor Courtney Noe
Senior Editor Jim Reed
Technical Editor Danny Humphress
Assistant Editor Susan A. Remini
Submissions Editor Jutta Kapfhammer
Editorial Assistants Monica Dorth, Valarie Edwards, Wendy Falk, Suzanne Benish Kurowsky, Lynn Miller, Shirley Morgan, Noreen Morrison, Kevin Nickols, Tamara Solley
Contributing Editors Dan Downard, Robert Frowenfeld, Randy Graham, Jim Hawk, Richard White
Technical Assistant Ed Ellers
Art Director Jerry McKiernan
Assistant Art Director Sally Nichols
Designers Peggy Henry, Neal C. Lauron, Kevin Quiggins
Advertising Manager Charlotte Ford (502) 228-4492
Advertising Assistant Debbie Baxter
General Manager Patricia H. Hirsch
Assistant General Manager for Finance Donna Shuck
Bookkeeper Diane Moore
Advertising Accounts Doris Taylor
Dealer Accounts Judy Quashnock
Administrative Assistant To The Publisher Marianne Booth
Director of Fulfillment Services Bonnie Shepard
Assistant Customer Service Manager Deidra Henry
Customer Service Representatives Sandy Apple, Monica Wheat
Word Processor Manager Lynda Wilson
Research Assistants Laurie Falk, Loretta Varda, Kara Voit,
Dispatch Mark Herndon
Production Assistant Melba Smith

Advertising and Marketing Office for the Western states and provinces: Cindy Shackleford, director, 12110 Meridian South, Suite 8, P.O. Box 73-578, Puyallup, WA 98373-0578. Phone: (206) 848-7766. Territories included: AK, AZ, CA, CO, HI, ID, MT, NV, NM, OR, UT, WA, WY, Canadian Provinces of Alberta, British Columbia, Saskatchewan.

PCM is represented in the Eastern United States by Garland Associates, Inc., P.O. Box 314, S.H.S., Duxbury, MA 02331, (617) 934-6464 or 934-6546. Advertisers east of the Mississippi may contact them for further information. Territories included: AL, CT, DE, DC, FL, GA, IL, IN, KY, ME, MD, MA, MI, MS, NC, NH, NJ, NY, OH, PA, RI, SC, TN, VA, VT, WV, WI, Canadian Provinces of Ontario, Quebec.

PCM — The Magazine for Professional Computing Management is published every month of the year by FALSOFT, INC., 9529 U.S. Highway 42, P.O. Box 385, Prospect, KY, 40059. Phone (502) 228-4492. **PCM — The Magazine for Professional Computing Management** and the **PCM** logotypes are registered ® trademarks of FALSOFT, Inc.

Second class postage paid Prospect, KY, and additional offices. USPS N 713-470 (ISSN 0747-0460). POSTMASTER: Send address changes to PCM, P.O. Box 385, Prospect, KY 40059. Forwarding Postage Guaranteed.

Subscriptions to **PCM — The Magazine for Professional Computing Management** are $28 per year in the United States. Canadian and Mexican rates are U.S. $35. Surface mail to other countries is U.S. $64, air mail U.S. $85. All subscriptions begin with the next available issue.

Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in United States currency only.

Limited back issues are available. Please see notice for issues which are in print and costs. Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in U.S. currency only.

Contents

Features

Departments

Advertisers Index

Cover illustration by Fred Crawford copyright © 1984 by Falsoft, Inc.

New Software Has 'Quality Of Excitement'

There is some real excitement around the PCM offices these days. And that excitement means some good things for you, too.

It seems like almost every day brings another piece of quality software for us to review and about which we can report to you. Most of this is for the Tandy 2000 — but we are seeing some top-flight programs for the Portable Computer, too. Naturally, in the coming months, you will see formal reviews on much of this software, in an effort to guide you in purchasing what you may need.

As we have said practically every time we have had the opportunity, we see the 2000 and the Portable Computer as being part of a "team." You can use the Portable while you are "on the road," (is that why we have a column by that name?) and then download information and the like to your 2000 when you reach home base. We think this offers the most flexible, least expensive and most powerful computing *system* available anywhere.

But what is really getting us excited is the quality of software that is becoming available for the 2000. And, because almost every bit of it allows for the transfer of files in and out, it should be a simple procedure to use the information — while, perhaps, not the programs themselves — in your Tandy "system."

I have been spending quite a bit of time with SPI's *Open Access* in the last couple of days. (You will find my review of this excellent software package in our reviews section in this issue.) And, the day after *Open Access* appeared, a little gem called *Lotus 1-2-3* made its presence known. Then, two days later, in walked Sorcim's *SuperCalc 3*, followed by Condor's *Condor 3*. We've also seen *Data Ace*, Telexpress' *Teleterm*, Micropro's *Professional* and a host of others. Some of these programs are available direct from Radio Shack, others available through the Express Order Software program.

Having looked at *1-2-3* for a bit on am IBM PC, we took a quick tour on the 2000. What a difference! Just goes to show what can be done with a really top-flight computer like the 2000. The speed is much better, there are more colors and the colors are more vivid. If you know anyone with a PC, set up your 2000 beside it, and both of you run *1-2-3*. There is no comparison! Which means, of course, there is no comparison between computers, either. But we've known that all along, right?

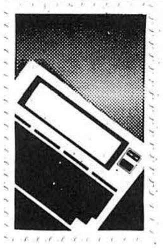
All this means you'll be seeing more information and more reviews in the coming issues of PCM, as well as more information about other aspects of your 2000 and Portable Computer. And, of course, we will continue to offer you original programs to perform a variety of tasks as well as fun things with both computers.

Yes, both the Portable Computer and 2000 markets continue to expand. You can help us by mentioning PCM when you buy or inquire about products — either direct from the manufacturer or through Radio Shack.

The fact that you see increases in advertising means that some of you are already helping us help you. Thanks for your support!

— Lonnie Falk

Letters



BEHIND BARS

Editor:

I have a problem with my bar code operation and wonder if others have experienced the same.

Specifically, wherever the program shows a "." (either decimal or period), the bar code reader prints an "!". Further, where a program calls for the letter 'Z', the bar code reader prints a "-".

For example, in Line 3060 of *Baread* ("Revised Reader," Page 26 PCM May '84) should read *3060 PRINT@280+ (20-.5*LEN)),ER$;* but the reader prints *(20- !5*LEN"* etc. In *Square* ("Square Deal," Page 23 PCM June '84), Line 100 should be *100 DEFINT A-Z:* But the reader prints *DEFINT A- :* and in Lines 615 and 625 it should be *:Z=1: and :Z=0:*, but the bareader reads it as *:-=1: and :-=0:*.

I have carefully checked the *Baread* listing on Page 18, PCM April '84, against that entered in my Model 100 and there are no differences. I mention this because it is the program I used to enter the modified *Baread* from the May issue of PCM.

I suspect the problem is in the B3OF9.C0, but since this is a machine program I don't know how to check it.

Can you offer any suggestion?

K.W. Klages
Orlando, FL

Editor's Note: You're absolutely right! The problem is not with the BAREAD .BA *reader program, but rather with the program that we use to print the code. The problem has been corrected and I have been sentenced to life "behind bars" by a jury of my peers on the* PCM *editorial staff.*

BCR INTERFACE

Editor:

I am the proud owner of a new M100 and spend a great deal of time with this machine. I am interested in interfacing the BCR socket to some electronic projects but have been unable to find any information on how the M100 inputs information through it. Can you help me? I am also anxious to see more articles on machine language programming in your magazine.

Martin Childs
Deep River, Ontario

Editor's Note: The 100's BCR port is designed to match the Hewlett-Packard bar code wand sold by Radio Shack and Bi-Tech; it's not well-suited to other applications.

AN EVIL GENIUS, MAYBE

Editor:

I recently bought a Disk/Video Interface for my Portable Computer. The computer is wonderful. The operating system is easy for me to use since I was already familiar with the bulk of it through the M100's operating system and the disk system is perfect for me in my job as a law clerk and as a student at the University of Kentucky. There is only one small problem in this whole scenario: the plug at the computer end of the D/VI cable. To say that the plug is not "user friendly" would not be doing it justice. It can only be attributed to some evil genius somewhere in the depths of Tandy or perhaps in Japan. Why such a plug was put on a *portable* computer of this type is beyond me, and it falls to me only to suffer with it or try to change it. Not being the martyr type, I would like to change it.

I have consulted preliminarily with some of my friends in the electrical engineering field and they are of the opinion that perhaps an easier plug could be spliced into the cable at a point near the computer, in effect making another connector just under the RS-232C port. My question to you is whether there is another cable or plug available to connect the M100 to the D/VI or, if not, is it possible to have a technician safely splice another, simpler connection into the line somewhere?

Any information on this problem would be greatly appreciated.

Matthew L. Mooney
Lexington, KY

Editor's Note: The only 40-pin connector we know of would be a pair of standard 40-pin edge connectors (0.1-inch pin spacing) and a two-sided card edge to mate the two; you would cut the cable about an inch from the back of the computer and put one of the connectors on each side, then insert the card edge in one and mate the two connectors. Check with an electronic parts distributor.

DOS IT RUN ON 2000?

Editor:

I have some software provided by an insurance company I represent and they had it programmed in PC-DOS, instead of MS-DOS. I would like to know if anyone produces add-on boards or whatever it would take to enable PC-DOS to run on the Tandy 2000?

I really like the 2000 and don't want to get rid of it for an IBM clone. I sure don't like the way they work as well as the 2000.

Charles LeBlanc
Brazoria, TX

Editor's Note: There is really no need for an add-on board for the Tandy 2000 to allow it to run PC-DOS. PC-DOS is what IBM calls its version of MS-DOS. There are very few differences between IBM's PC-DOS and MS-DOS — most of which do not directly affect program compatibility. Most PC-DOS programs need only be transferred to a Tandy 2000 MS-DOS disk in order to run. This is a simple process since the 2000 can read from and write to PC-DOS disks. A few programs that directly access IBM PC hardware, or make extensive use of IBM graphics, will have to be revised for the 2000. The best way to find out if your package will work on the 2000 is to give it a try. Just copy it (using the MS-DOS COPY command) to a backup of the MS-DOS disk that came with your computer.

TWO FOR THE COMMENTS

Editor:

I would like to thank you and your writing staff for an informative Model 100 magazine. I follow PCM with the same enthusiasm as your sister publication RAINBOW (which nicely complements my CoCo).

I wanted to comment on a few changes occurring in PCM which I'd like to hear from other readers:

1) The bar code reader programs are just fantastic. I picked up a Radio Shack bar code reader at the Dayton amateur radio convention and it has already paid for itself in the time saved on typing in programs. Keep up the good work.

2) I'm a little disappointed in seeing PCM pick up the Tandy 2000 computer for inclusion in the magazine. I won't tell you your business, but I think your magazine can exclusively support the breed of portable computers out now and the second generation that will come from Tandy, NEC, etc. You mentioned in your editorial that the M100 and the Tandy 2000 are natural partners. Well, it's hard for me to see how much more "natural" the 2000 is, as compared to the IBM PC, Apple, or in that case the CoCo (I am able to upload and download into my CoCo disk system as easy as any other system). Maybe what needs to be done is to take a reader survey and find out what "other" computers are being used (if any). I'll bet you'll find the hobbiests with CoCos, Commodore 64s, Apples, etc. and the business users with IBM PCs and clones (just step in any large company office and see how many PCs and Apples are in use).

The M100 is really a true stand-alone unit that offers tremendous utility with *any* other computer. Please don't dilute your fine magazine with the Tandy 2000. If you really think it will succeed (I have my doubts), do what you were maybe planning — start a dedicated Tandy 2000 magazine.

Let's have some reader input on this subject. Thanks again for a nice publication.

Bob Gobrick
Lakewood, CO

Ask your dealer for the Porta™ series from Skyline—to help you work faster and better!

PortaCalc™ — Model 100 "electronic spreadsheet" with 14 column by 26 row workspace! Full use of the built-in function keys to save, load, screen print, report print, or to look behind the data at the formulas in use. PCM

Full arithmetic operators including exponentiation, absolute value, integer, summation, and averaging. Calculations are done to 14 digits of precision and displayed up to 9 digits. User selectable decimal place from none through seven—not just one, two, or floating! Formula replication is included to allow fast creation of worksheets.

Worksheets may be saved, loaded, or merged using the computer's memory or cassette. Often used templates can be saved in memory for instant recall.

Two powerful utility programs are included with **PortaCalc** at no extra charge! **PortaDex**™ is a data exchange program that allows reformatting **PortaCalc** files into the DIF format used by **VisiCalc**. **PortaPrint**™ is an enhancement to the Model 100's text editor, adding the ability to adjust left, right, and top margins, and page length. Page numbering, headers, centered lines, flush right justification, new page control, and more.

Comes with extensive documentation in padded 3-ring binder. Includes tutorial, detailed reference section, and executive level sample templates. Fully illustrated with screen prints, examples, and hints.

PortaStat™ — Correlation, regression analysis, and descriptive statistics package, designed to be a working tool for today's businessman. Data can be input from cassette, memory, or keyboard. Interactive with **PortaCalc** data files! Thoroughly documented, includes tutorial.

PortaFin™ —Present value, net present value, future value, annuities, interest factors, loan constants, internal rate of return and more! This user-friendly package prompts you for the needed input data, then gives you the answer to your loan or investment problem! Interactive with **PortaCalc** files, allowing you to "read" cash flow data from memory. Keyboard or cassette input also available.

PortaMax™ — The powerful 'simplex' method of linear programming. Find the optimum 'mix' for a given set of constraints. Interacts with **PortaCalc** files to avoid time-wasting repetitive data entry. Tutorial, examples, and bibliography included.

PortaFolio™ — Stock and bond valuation, betas, Macaulay's duration, yield to maturity, bond swap calculations, and much more. All in an easy to use program, interactive with **PortaCalc** data files. Supplied with comprehensive instructions and reference manual.

PortaCalc

All **Porta** series programs are supplied on cassette and require 24K RAM.
Pricing — PortaCalc is **$69.95,** all others are **$44.95** each.
If ordering direct, add $2.00 shipping and handling per order.
Dealer inquires invited.

VISA MasterCard

Model 100 is a trademark of Tandy Corp VisiCalc is a trademark of VisiCorp DIF is a trademark of Software Arts

Skyline
MARKETING CORP

4510 W. Irving Park Rd. • Chicago, IL 60641 • (312) 286-0762

Basic Bytes

Dealing With Basic String Functions

By Richard A. White
PCM Contributing Editor

We think of computing as dealing with numbers for the most part. There are some programming languages whose reason to be is to deal with numbers. While dealing with numbers is important, dealing with characters and words has become a major task for modern computers. Word processing and database management, including mailing list data, client listings, employee information listings and inventory work depend on computerized handling of strings.

A string is a series of characters. Characters are anything not treated as a number. Letters, numerals, punctuation, control characters such as carriage return and back space and graphics characters are all just characters and are treated in the same way by BASIC. A string of characters does not have to make sense to still be a character string.

Characters can be entered into the computer in a number of ways. We will concentrate on program and keyboard entry, leaving data entry from cassette or disk to other columns. Strings may be part of a program. Here are two examples:

```
10 S$=“DEMONSTRATION”
20 PRINT“THIS IS A ”S$
```

In Line 10, the string “DEMONSTRATION” is assigned to the string variable S$. As we have said before, S$ does not equal the string. It represents the string and allows the computer to find the string. Once BASIC runs Line 10 it knows where S$ string is and can use it anywhere in the program. There is also a string in Line 20, “THIS IS A ”, that is not assigned to a variable. This is a string that BASIC uses when it comes to it in Line 20 and nowhere else. Note that BASIC also uses S$ in Line 20 to print THIS IS A DEMONSTRATION.

Strings can also be put in a program in DATA statements. In this case, the DATA must be read into string variables before it can be used.

```
10 FOR X=1 TO 4 : READ S$(X):
S$(X)=S$(X)+“ ”:NEXT
20 FOR X=1 TO 4 : PRINT S$(X);):
NEXT
30 END
1000 DATA THIS, IS, A,
DEMONSTRATION
```

A FOR . . . TO . . . NEXT loop reads the DATA into S$(X) string variables. We then add a space to each string. A FOR . . . TO . . . NEXT loop is used to print the strings and we get the same THIS IS A DEMONSTRATION printout we had before. Note that the program never reached Line 1000, but found the DATA and printed it all the same. BASIC is set up to find the DATA wherever it is put in the program. The obvious place for DATA statements is at the very end of the program where

(Richard White has a long background with microcomputers and specializes in BASIC *programming. With Don Dollberg, he is the author of the* TIMS *database management program.)*

BASIC can find the information, but does not waste time running over DATA lines.

I prefer to define strings at the end of a program, send the program over these lines once and have them available from then on. Here is an example:

```
5 GOSUB2000 : ...
2000 S1$=“MESSAGE 1” : S2$=
“MESSAGE 2” : S3$=“MESSAGE
3” : RETURN
```

INPUT A$, LINEINPUT A$ and I$= INKEY$ all seek a keyboard input that is put into a string. INPUT A$ accepts characters from the keyboard until it sees a carriage return (ENTER), quote, comma, or semicolon. This is good news and bad news. The good news is that you can enter a number of strings with one INPUT statement. The bad news is the obvious limit on what the string you enter can contain. INPUT also lets you print a string to the screen telling the operator what to enter.

```
100 INPUT “ENTER YOUR NAME
AND TELEPHONE NUMBER”;
NA$,TL$
```

LINEINPUT A$ allows entry of any keyboard character except the carriage return (ENTER) which terminates the string. It allows only one string to be entered at a time and does not support printing a string after the keywords. The equivalent of the above INPUT code follows.

```
100 PRINT “ENTER YOUR NAME
AND TELEPHONE NUMBER” :
LINEINPUT NA$ : LINEINPUT
TL$
```

The big value of LINEINPUT is in wordprocessing and database management programming where all keyboard characters need to be entered into strings.

In the ideal program, all operator choices will be entered as single keystrokes. Why enter a character and an ENTER when we can do the same thing with one keystroke. INKEY$ is our hero.

```
20 I$=INKEY$ : IF I$=“” THEN 20
ELSE RETURN
```

This subroutine is a real workhorse. Put it early in the program so BASIC can get to it in a hurry. You can then call it from various places in the program.

Once the character or string is entered, it is processed or used in some way. BASIC includes a variety of powerful processing facilities. Strings can be combined or added to each other in a process called concatenation.

```
5 GOSUB 2000
100 A$ = B$ + C$ + D$ : PRINT A$ :
END
2000 B$=“THIS ” : C$=“DEMOS ” :
D$=“CONCATENATION.” :
RETURN
```

Here three strings were added to form a new one A$ which was then printed. This becomes a powerful tool when you need to make a string that includes characters not available from the keyboard. There is no other way to make a string in a program that contains a quote than to add it to a string CHR$ (34). How would you write BASIC that prints the following line exactly as shown onto the screen?

```
10 A$ = “TEST STRING”
```

First try might be:

```
10 PRINT “10 A$ = “TEST
STRING””
```

But, that won't work since BASIC will read the second quotation mark as the end of a string. It will think that TEST

STRING is a variable, which has not been evaluated, of course, and print a zero. The final "" is thought of as a null string, a string with nothing in it, which prints nothing. If you run Line 10 you get:

```
10 A$= 0
```

The ASCII value of the quotation mark is 34. Let's rewrite Line 10 to tell the program to print CHR$(34), the character whose ASCII value is 34 whenever we need to print a quotation mark. We will put the whole thing into a new string so we can easily print it any time we need.

```
5 A1$ = “10 A$= ” + CHR$(34) +
“TEST STRING” + CHR$(34)
10 PRINT A1$
```

This will do the job. Key it into your M100 and prove it to yourself.

What your Model 100 can put together, your Model 100 can take apart. Taking strings apart, or getting pieces of strings, is fully as important as putting strings together. M100 BASIC offers three functions that return a part of a string.

```
L$ = LEFT$ (A$,NUMBER)
M$ = MID$ (A$,POSITION,
        NUMBER)
R$ = RIGHT$ (A$, NUMBER)
```

In all cases, A$ is the string we need to get something from or take apart. NUMBER is simply the number of characters from the left end of A$ that need to be put into L$. A$ remains unchanged, and the characters in question are in both A$ and in L$. When RIGHT$ is used it works the same way except it counts back from the right end of A$ to determine which characters to get.

```
L$ = LEFT$ (“JULY, 1984”, 4)
L$ will return “JULY”
R$ = RIGHT$ (“JULY, 1984”, 4)
R$ will return “1984”
```

MID$ is more general, and more powerful, but has only one more parameter, the starting position in the string, POSITION to remember. POSITION is the number of characters from the left end of the string where BASIC is to start getting characters. NUMBER is the length of the string to get and put into M$. Did you happen to think that we can throw LEFT$ and RIGHT$ away and use MID$ for all string disassembly? The substitution for LEFT$ is simply to use 1 for position.

```
L$ = MID$ (A$, 1, NUMBER)
```

This will work exactly like LEFT$ above.

```
L$ = MID$ (“JULY, 1984”, 1, 4)
L$ will return “JULY”
```

To substitute MID$ for RIGHT$, we need to know how long A$ is and subtract NUMBER from that length plus 1 to get the starting position.

```
100 L = LEN(A$) : R$ = MID$(A$,
L+1-NUMBER, NUMBER)
```

Let's reason it through.

```
A$ = “JULY, 1984)
L = LEN(A$) L will return 10
NUMBER = 4
R$ = MID$(A$, 10+1-4, 4)
R$ will return “1984”
```

Looks simple, doesn't it? Well, it is and it isn't. The problem is not in knowing how to use LEFT$, RIGHT$ and MID$. The problem comes when you need to write the program to calculate POSITION and NUMBER. This depends on the logic of the program you are writing. String handling functions are basically simple. It's how they are used that gets complicated.

A powerful help is INSTR. Its syntax is:

```
N = INSTR(NUMBER, SEARCH
STRING, TARGET STRING).
```

SEARCH STRING is the string to be searched. Within that string we hope to find the shorter TARGET STRING." We will start at character position NUMBER, counted from left to right. If we get a match, N will carry the number of characters from the left that the TARGET STRING starts. If the TARGET STRING is not found, N = 0.

First, this is a powerful comparison utility. Say you had an inventory file and you wanted to find WIGITS, LH. Let's assume that names of things were in the NA$() array. If we can find the array member number for WIGITS, LH, then we can access the data relating to WIGITS, LH that would be in other arrays.

```
200 PRINT“PART NAME”:
LINEINPUT PN$
210 FOR X = 1 TO LT : N=INSTR
(1,NA$(x),PN$)
220 IFN=0 THEN NEXT ELSE
PRINTNA(X) “  ”IV(X)“  ”CO(X)
“  ”PR(X)
```

Here LT is the number of entries in the inventory file. The user would type in WIGIT, LH. Since this contains a comma, we must PRINT the prompt and use LINEINPUT. Now I would expect only one NA$() to contain WIGIT, LH, so N = 0 until the record is found. Then N will probably return 1 since we expect the name to start in the first position in the variable, but it does not have to. Anyway, N will be greater than zero, routing the program to the print statement after the ELSE. In the example this material printed, would be: name, NA$(), amount in inventory, IV(), cost of each CO(X), and the selling price, PR(X).

There are a variety of other uses for INSTR. I have used it to make a phrase substitution editor, for one. Another powerful use is as part of a program control strategy. Say, for instance, you have a menu where choices can be entered as single letters.

```
dARK BEER
lIGHT BEER
pLAIN PIZZA
hOT DOG
cOLA

KEY FIRST LETTER OF CHOICE
```

Only in Prospect, Ky., does an Eatery with such a limited choice stand to survive. The allowed entries will be one of five letters, but could be upper or lowercase so there are really 10 options. Sorting these out with IF . . . THEN . . . ELSE statements would be a drag and eat memory to boot. Let's use INSTR.

```
100 CLS:PRINT“ dARK BEER”:
PRINT“ lIGHT BEER”: PRINT
“ pLAIN PIZZA”: PRINT“hOT
DOG”: PRINT“ cOLA”: PRINT:
PRINT“ KEY FIRST LETTER OF
CHOICE”
110 N = INSTR(1,“dDlLpPhHcC”,
I$):
115 GOSUB20: ON N GOTO150,
150,200,200,250,250,300,300,
350,350
120 GOTO100
```

Line 100 prints the menu choices. In Line 110, we call the INKEY$ in Line 20. Then INSTR hunts through the SEARCH STRING for a match for the character in I$ and returns a number from one to 10 if there is a match. In Line 15 view the ON N GOTO as counting across its list of line numbers *to the* Nth one and going to that line. If no match is found, the program falls through to 120 and goes back to 100 to try again.

The final function to mention this month is STR$(). Its syntax is A$ = STR$(X). It just converts the number in X to a string that has a leading space.

The Executive Decoder Ring

A Different Approach To File Security

By J. Kenneth Guscott

While many might envision the coding and decoding of messages as indulging in a fantasy world of spies and international intrigue, it is, in fact, a practical method of maintaining file security during data communication, as well as being a means of keeping sensitive RAM files private.

(Ken Guscott is an engineer engaged in the design of electronic and optical instruments. He holds a bachelor's degree in electrical engineering and designs software primarily for engineering and mathematics applications.)

I suppose that everyone has a little James Bond in them, and has at some time or another fooled around with codes and ciphers. When I was in high school some of us used to correspond with coded messages based on a letter substitution method.

It was great fun and had an air of mystery about it, although all of our codes were quite simple to break. It was about that time, that I first discovered the "Alphabet Cipher" which was invented by the Reverend Charles Lutwidge Dodgson, a mathematical lecturer of Christ Church College in England. "Who's he?" you ask. Well, his pen name was Lewis Carroll and he wrote *Alice In Wonderland* and a lot of other things including books on symbolic logic. Now here was a great code, and one extremely difficult to break because it is based on a "Key" which is known only to the correspondents. This method has been used by spies through generations and has been featured in many spy novels and movies. In these stories the spies usually had copies of the same innocent looking paperback book which was used as the key text for encoding and decoding vital messages.

Well, now we can take our innocent looking Model 100 and turn it into the fabulous "Lecturn" cryptograph machine which gave James Bond so much trouble in "From Russia With Love." With any luck a beautiful spy will come out of the woodwork and our adventures will begin. We will develop a powerful cipher program which, at the same time, will demonstrate how to use the files protocol in the Model 100 interactively. This is a most useful concept for many program applications.

First, let us review how the book code works. Figure 1 shows a matrix of alphabets which is used to encode and decode the message. Let us suppose that our KEY text is "MARY HAD A LITTLE LAMB." The program repeats it as many times as necessary to encode the message. Now let us suppose our message is "MEET ME BEHIND THE BARN AT MIDNIGHT AND WE WILL WATCH THE MOON TOGETHER." This will be encoded as follows.

```
MARY HA DALITT LEL AMBM
AR YHADALIT TLE LA MBMA
RYHAD ALI TTLE LAMBMARY
MEET ME BEHIND THE BARN
AT MIDNIGHT AND WE WILL
WATCH THE MOON TOGETHER
```

Read the key text into the top horizontal row of Figure 1. The first letter is 'M,' so go into column M. The first letter of the message is also M so go to row M. The letter at the intersection is the first letter of the encoded message. Keep doing this and the encoded message will read:

ZFWS UF FFTRHX FMQ CNTA
BL LQERJSQN UQI IF JKYM
OZBDL UTN GIAS FPTGGIWQ

Of course, any worthwhile spy would probably try to make this even harder to crack by sending the message as:

ZFWSUFFFTRHXFMQCNTABL
LQERJSQNUQIIFJKYMOZBDL
UTNGIASFPTGGIWQ

This way we don't even know how many words there are or how many letters the words have. To decode the message the table is used backwards, so to speak. Go to the column of the letters of the key text in turn as before, and look down the columns until you see the corresponding letter in the encoded text. Look to the left column of that row to get the decoded letter.

FIGURE 1 — Alphabet Cipher

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
B	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
C	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
D	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
E	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
F	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
G	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
H	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
I	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
J	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
K	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
L	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
M	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
N	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
O	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
P	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
Q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
R	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
S	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
T	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
U	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
V	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
W	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
X	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
Y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
Z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Well, that's how it works, and you can see that it would be quite tedious to do this manually from the table as spies in the old days did. If you think about the operation of the code and attach a number to each letter, so that A = 1, B = 2, etc., then all that the table does when encoding is to add two letters together and so come up with a new letter. For example, A + A is 1 + 1 = 2 which is B. If the sum is higher than 26, then 26 must be subtracted from that number. For example, P + Q is 16 + 17 = 33, 33 - 26 = 7 and 7 is G. This gives us a clue on designing a suitable algorithm for Lewis Carroll's code. It will be something like this:

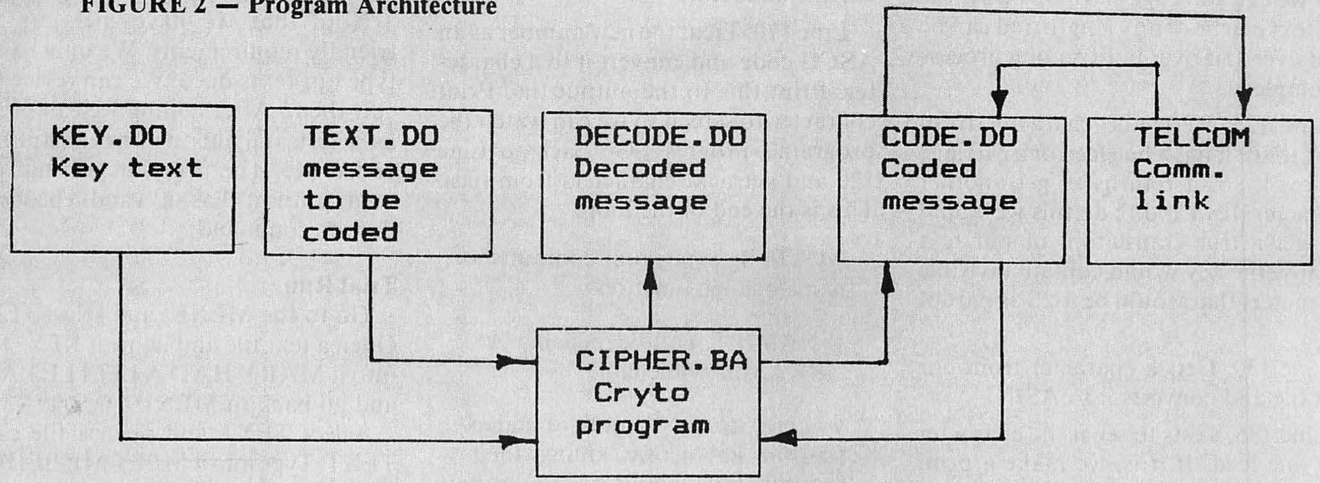
```
ENCODE
Get the first characters from Key and
Message
Convert letters to numbers
Code = Message + Key
If answer is greater than 26, subtract
26
Convert code number to code letter
Get the next pair of characters

DECODE
Get the first characters from Key and
Code
Convert letters to numbers
Message = Code - Key
If answer is less than 1, add 26
Convert message number to message
letter
Get the next pair of characters
```

Fortunately, this task is very easy for our computer, so we can greatly improve on the method. In the discussion above we are only using the 26 letters of the alphabet. With the computer we can expand the table to employ both upper- and lowercase alphabets, the numbers and even all of the punctuation and symbols that are at our disposal. In the example given above, after decoding we would have to figure out the message and put in the necessary spaces and punctuation. This could lead to some ambiguities in real life, I'm sure. In the case of the computer we can encode everything, including spaces and punctuation and the code will be a continuous flow of symbols. In the computer all of our alphabets and symbols already have numbers assigned to them and these are, of course, the ASCII code. It is a very simple matter to write a program to encode and decode our messages for us using these numbers.

How hard is this cipher to crack? Nearly impossible. Consider the method of encryption, two disassociated characters are added together to get a sum. Without knowledge of the key there is no way that the coded text means anything, or has any pattern to it. It is similar to getting the result of an arithmetic addition. If you are given that the sum of two numbers is 10, for instance, the only thing you can say is that the original two numbers could be 9 + 1, 8 + 2, 7 + 3, 6 + 4, or 5 + 5, assuming they were integers. In the case of our cipher we are given the equivalent that the sum of two numbers lies between 32 and 122. It would require a lot of computer aided substitution to decode it, because there is no discernible pattern. You might get somewhere if you knew that the key was a repeating group of letters, but in our case we can make the key a 300-page novel and never use the same passage twice. If our message had 100 characters in it we would have to make 90 to the power 100 substitutions to get all combinations.

If we could make a million substitutions per second in one year this would only yield 3.1536E13, and at this rate it would take quite a few lifetimes. You may ask "If it's so good and so simple, why isn't it used by government and commercial interests?" How do you know it isn't? Seriously, though, the problem is in the portability of the key. It's okay for a few individuals to use, but how would you issue and update the key throughout an army, for instance, and make sure that everyone knew what passages of text to use today? How would you ensure the key did not fall into enemy hands? Once it did, computer substitution becomes relatively easy. Just start at the beginning of the book and decode, then move one character, do it again, and so on. If the book has 200,000 characters, that's just 200,000

FIGURE 2 — Program Architecture



tries. Once you had entered the key text, you could decode quite quickly. Of course, it would take a big mainframe to do it and it would have to have a program that could recognize character groups as language.

Program Architecture

Whenever we start to write a program, good practice advises that we first consider the output we want and the input necessary to achieve it. If we cannot define these exactly, we have no business slinging code. In this case the output is simple enough: we want a file containing the encoded message, or one containing the decoded one. For input we must have the key text, and the message either in clear text or in code. The term "user friendly" is heard a great deal lately, which means that the program must be designed so that you don't have to type an input over if you make a mistake. These objectives will be part of our program requirements.

It turns out that this is very easy using the Model 100's excellent filing system. If the key text is written in a file, and the message and output are similarly treated, then these can interact with the cipher program with a minimum of hassle. With this arrangement we can edit any file without having to re-enter the whole thing. Figure 2 shows the general arrangement for our program.

Now, let's take a look at the cipher algorithm. If we look at the ASCII code table for the Model 100 we find that we will not want to use the whole table. Space, ASCII 32, is the lowest code of interest because we want to be able to encode spaces, and lowercase z is the highest at ASCII 122. We don't want to use the codes below 32 because a lot of these are control commands for the computer and peripherals. The ASCII codes above 128 are also taboo, since there is very little standardization on what they should represent. Some computers don't even recognize them at all and we want to achieve compatibility with any computer.

If we select 32 and 122 as the limits, our coding algorithm will take the following form.

```
ENCODING
Get the first characters from KEY and TEXT
Convert character to its ASCII code
ASCII CODE = ASCII KEY + ASCII TEXT
If ASCII CODE is greater than 122 subtract 90 (122 - 32). Repeat if necessary.
Convert ASCII CODE to character and print to file
Get next pair of characters

DECODING
Get the first characters from KEY and CODE
Convert character to its ASCII code
ASCII DECODE = ASCII TEXT - ASCII KEY
If ASCII Code is less than 32 add 90.
Repeat if necessary.
Convert ASCII Code to character and print to file
Get next pair of characters
```

This sounds good, but there is a problem. Suppose when we code we have a lowercase z (122) in the TEXT or KEY. This gets added to the other. Let's assume '50' as an example.

122 + 50 = 172 172 - 90 = 82

This will be the coded character. When we decode we will get:

82 - 50 = 32

Horrors, our z has become a space and this is true of any number added to 122 — the result will be the same. Similarly, we can get some confusion in the decode algorithm. If the code is 122 or 32 this happens.

122 - 50 = 72 32 - 50 = -18 -18 + 90 = 72

Thus z and space in the coded text have been decoded as the same character. This will never do and the easiest way to fix it is to allow the program to use the ASCII numbers 31 and 123 which is permissible because the characters they represent will never appear in our message text. The correction factor then becomes 92.

Walk Through The Program

Line 100: Clear screen, define all variables as integers (faster operation), declare number of files and open the file KEY which should contain our key text.

Line 105: Determines if we wish to code or decode text and tests for a valid input. As part of the making the program "friendly," we will accept any word starting with the letters "E" or "D" upper- or lowercase as valid. For anything else we get reprompted.

Line 110: Opens an appropriately labeled file for our output. We use CODE if we are encoding, otherwise DECODE.

Line 115: Display all files in the computer so that we can select the one we want to work on. TEXT is a good title for the message if you like, but we can work with any document file.

Line 120: Start of encryption loop. Gets one character at a time from the KEY file and converts it to its ASCII equivalent. Tests for end of file and if found closes KEY file and re-opens it so that we can start again. In this way, the key text can be of any length and can be used over and over until coding process is complete.

Line 125: Tests the character from KEY to see if it is a line feed or a carriage return. If so, it returns to get another character. If we didn't do this we would not get a true translation of our text because the key would contain invisible characters that would be used as part of it.

Line 130: Gets a character from our text file and converts it to ASCII.

Line 135: Tests to see if the character is a line feed. If it is, we make a print statement to our output file so that it will end its print line the same way as the text file. It then tests for end of file and if so, exits to the finish statements. If not, it gets another character.

Line 140: Tests if character is a carriage return. If so it gets another character. We don't want extra carriage returns or we will get a double-spaced output. When we send the print statement in Line 135, it will give us both line feed and carriage return.

Line 145: Test for end of text file. If so exit loop to finish statements.

Line 150: If we are encoding we sum the ASCII codes of the KEY and TEXT characters.

Line 155: Test the summation to see if the result is within the valid range. If not, subtract the correction factor. The correction is applied as many times as necessary.

Line 160: If we are decoding we subtract the KEY from the TEXT.

Line 165: Test the result to see if it is within the valid range. If not, add the correction factor.

Line 170: Treat the new number as an ASCII code and convert it to a character. Print this to the output file. Print character to screen so we can watch the program's progress. Go back to Line 120 and get next characters from files. This is the end of the loop.

Line 175: Finish statements tell us coding is complete and where filed.

Line 180: Delay loop to allow screen to be read, and then return to menu so that we can access our files.

Line 185: Subroutine for testing for valid encode input.

Line 190: Subroutine for testing for valid decode input.

Line 195: Subroutine to indicate input is not valid and needs to be reprompted.

This is a very simple program, but it demonstrates very nicely how the Model 100 files and programs can interact. Now that we have our encoded message in *CODE.DO*, we can go to *TELCOM* and transmit it to the recipient, or leave it on a bulletin board for pick up. This is another example of interaction.

> NOTE: By using your computer's name-change function
>
> NAME"RAM:old filename"AS "RAM:new filename"
>
> you can store a number of coded files in RAM. By killing their straight-text counterparts and either killing the *Cipher* program and/or *Key* text file (or making them invisible to the MENU function as published in the February 1984 issue of PCM) you can maintain strict file security.

Note that we have met our user friendly requirements. We only have to type our texts once. We can review and edit them. After coding we can decode as a test and this will not disturb the coded file. The program will not proceed until it has a valid encode or decode command.

Trial Run

Go to the MENU, and select TEXT. Open a text file and name it KEY. Type into it MARY HAD A LITTLE LAMB and go back to MENU.

Select TEXT and open a file called TEXT. Type into it MEET ME BEHIND THE BARN AT MIDNIGHT AND WE WILL WATCH THE MOON TOGETHER and go back to the MENU.

Select CIPHER and enter E when prompted. Type in TEXT as a response to the file request and the cipher will appear on the screen. If you did it right the code will read:

```
>*;Q@9*db*h9;<tD1el'28?a7Q@9.
,n.g8At9>-@G*m=:1Byw-9++hat82
tE?8nl9@-69>Br
```

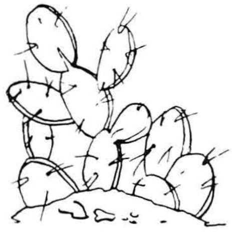
You can now repeat the procedure selecting D and CODE and you will see the decoded message.

Okay, kindred secret agents, let's go to work. SPECTRE, watch your step and SMERSH, eat your heart out, our messages are now secure. One final hint. When you go out and buy 15 copies of your romance novel to use as a key, don't buy them all at the same time or someone might get suspicious. Also, make sure they are all the same edition, we don't want any slip-ups.

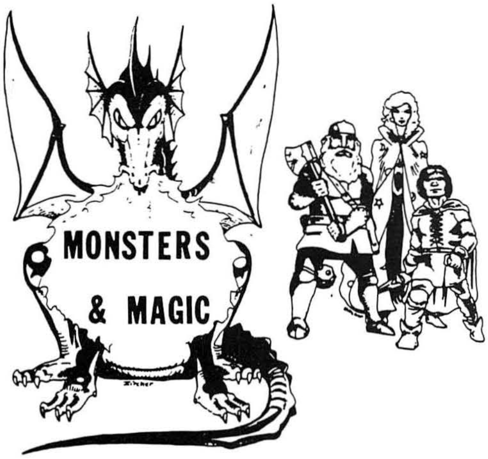
PCM BAR CODED LISTING

The listing:

```
100 CLS:DEFINTA-Z:CLEAR:MAXFILES=3:OPEN"
KEY"FOR INPUT AS 1
105 INPUT"(E)ncode or (D)ecode ";C$:GOSU
B 185:IF C$=""THEN 105
110 IF C$="E"THEN OPEN"CODE"FOR OUTPUT A
S 3 ELSE OPEN "DECODE"FOR OUTPUT AS 3
115 FILES:INPUT"Select subject file ";F$
:OPEN F$ FOR INPUT AS 2
120 K$=INPUT$(1,1):K=ASC(K$):IF EOF(1) T
HEN CLOSE 1:OPEN"KEY"FOR INPUT AS 1:GOTO
 120
125 IF K=10 OR K=13 THEN 120
130 S$=INPUT$(1,2):S=ASC(S$)
135 IF S=10 THEN PRINT:PRINT#3,:IF EOF(2
)THEN 175 ELSE 130
140 IF S=13 THEN 130
145 IF EOF(2) THEN 175
150 IF C$="E"THEN O=S+K
155 IF O>123 THEN O=O-92:GOTO 155
160 IF C$="D"THEN O=S-K
165 IF O<31 THEN O=O+92:GOTO 165
170 O$=CHR$(O):PRINT O$;:PRINT#3,O$;:GOT
O 120
175 PRINT:IF C$="E"THEN PRINT"Coding com
plete, stored in file CODE.DO"ELSE PRINT
"Decoding complete, stored in file DECOD
E.DO
180 FOR I=1 TO 1000:NEXT:MENU
185 IF LEFT$(C$,1)="E"OR LEFT$(C$,1)="e"
 THEN C$="E":RETURN
190 IF LEFT$(C$,1)="D"OR LEFT$(C$,1)="d"
 THEN C$="D":RETURN
195 C$="":RETURN
```



PRICKLY-PEAR SOFTWARE
QUALITY PROGRAMS FOR YOUR MODEL 100

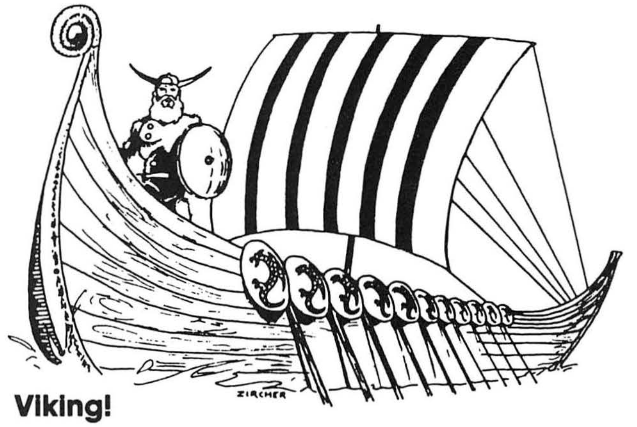


Monsters & Magic

The most realistic fantasy role-playing game yet for the Color Computer. You start out by rolling up your character's basic ability scores and buying equipment. When you are ready it's into the dungeon. Your character starts at first level, but can rise in levels by garnering experience in the fray. If you have role-played fantasy games you will be amazed at the realism of the combat system. Armor class, initiative, and damage by weapon type are all included, with over 50 different monsters to fight - each with it's own abilities. As you rise in level you can win treasure and find magic weapons and spells - if you live!! You set the game length by telling how many monsters you want to fight before you reach your final battle to the death against the powerful Dungeon Lord. There are 1000+ place description combinations in this text based game, and real excitement in every one! This is a fantasy simulation, and is truly not like any adventure game you have ever seen. For 1 player. **$19.95 tape—24K**

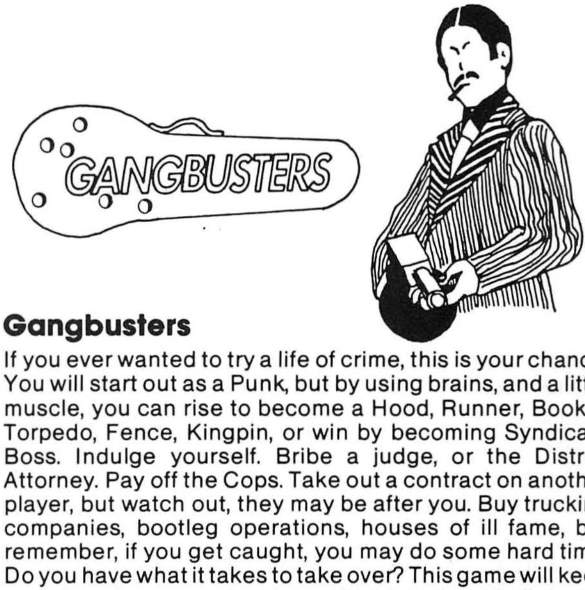
Astrology

Truly a classic, this program will accurately cast your complete horoscope. You just enter the date, time, and place of birth. The sun sign, rising sign, mid heaven (MC), lunar nodes, and planetary influences including houses and aspects between the planets will all be calculated, and a full chart drawn. You can also do progressed charts and transits. It will even tell you the day of the week you were born. The accompanying book will help you interpret this chart of your horoscope. The extent of the documentation is tremendous, even by our exceptionally high standards, and no previous knowledge of the subject is required. You can share in this wisdom which has been used for thousands of years in many cultures. This program was written by a professional Astrologer. **$34.95 tape – 16K**



Viking!

A simulation for 1 to 4 persons. Each begins as a landowner, and by farming their land, buying and selling land, expanding their fishing fleet, building on to their manufactory, increasing their population, equiping and training more soldiers, and regulating their taxes, each player tries to increase their economic power and rank until one becomes ruler over all. But beware plagues, rats, raiders, revolts, bad weather, and other misfortunes which may lie along the road to success. As you progress, see the map of your holdings increase. Playable in 1 to 2 hours, and different every time, you may have an addiction problem. **$19.95 tape — 24K**



Gangbusters

If you ever wanted to try a life of crime, this is your chance. You will start out as a Punk, but by using brains, and a little muscle, you can rise to become a Hood, Runner, Bookie, Torpedo, Fence, Kingpin, or win by becoming Syndicate Boss. Indulge yourself. Bribe a judge, or the District Attorney. Pay off the Cops. Take out a contract on another player, but watch out, they may be after you. Buy trucking companies, bootleg operations, houses of ill fame, but remember, if you get caught, you may do some hard time. Do you have what it takes to take over? This game will keep you close to your rod, get you thinking about bulletproof glass in your car, and definitely bring out the worst in you, but you'll love every minute of it. For 2 to 6 players, takes about 2 hours to play. Every game is excitingly different. **$19.95 tape — 24K**

Your personal check is welcome - no delay. Include $1.50 shipping for each program ordered. (Shipping free on $50.00 or larger orders). AZ residents add 7% sales tax. Orders shipped within two days.

Stocked by Quality Dealers, or

Send Order To: **PRICKLY-PEAR SOFTWARE**
9234 E. 30th Street
Tucson, Arizona 85710
(602) 886-1505

MasterCard VISA

Mastering MS-DOS

Part IV

By Danny Humphress
PCM Technical Editor

If you've followed this series for the past three months, you are already familiar with the fundamentals of our favorite operating system, MS-DOS. You learned about file and directories and you've begun to use commands that manipulate files. To date, the files that we've been working with have been those that were on our MS-DOS disk when we received it. This month, we're going to learn how to create files using a special feature of the COPY command and MS-DOS's built-in "mini-word-processor," EDLIN.

The Many Faces of COPY

We learned that COPY will make a copy of a single file or group of files from one disk to another. It will also allow you to have two copies of a file on the same disk as long a both copies have different names or are in different directories (more on directories later).

The great thing about COPY is that it's not limited to just files. Remember the very first MS-DOSsier when we discussed how a computer is a combination of *devices* such as a disk drive, keyboard, printer, and display. COPY's main function is to copy data from one device to another. Granted, the devices that COPY usually uses are the disk drives, but it can do much more than that. Take a look at this command line.

COPY A:PAYROLL.DAT B:

You'll remember from last month that this command will copy the file called "PAYROLL.DAT" from drive A: to drive B:. The 'A:' and 'B:' in this command line are device names for the two disk drives (refer to MS—DOSsier Part I). We could easily substitute just about any other device name for 'A:' or 'B:'. For example:

COPY A:PAYROLL.DAT LPT:

This command would copy "PAY ROLL.DAT" from drive A: to the line printer since "LPT:" is the device name for the line printer. The contents of the file would be printed by the printer. See if you can deduce what the following command would do:

COPY LPT:PAYROLL.DAT A:

Of course, it would copy "PAY ROLL.DAT" from the printer to drive A:. Wait a minute! You can't copy a file *from* a printer, the printer is an "output only" device. There are some cases, like that above, that are not logically valid, so think about what you're trying to do when using COPY this way.

You can only COPY from input devices to output devices. Some devices, such as the disk drives (A:, B:, C:) are considered "input/output" devices, or "I/O." and can be used for either input or output. Another example of an input/output device is the "console." The console (CON:) is the logical device name of the keyboard and display combination. Input from the console comes from the keyboard (of course), and output goes to the display screen.

Given what you know about the CON: device, determine what the following command line would produce:

COPY A:PAYROLL.DAT CON:

The contents of "PAYROLL.DAT" on drive A: would be "copied" to the display. In effect, the data in the file would be displayed on the screen. This can be very useful when you want to see what's contained in a file. Since this is something that is so often needed, the creators of MS-DOS provided us with an easier to use alternative to this with

(Danny Humphress, PCM's Technical Editor, is the owner of a computer software and consulting firm in Louisville, Ky. Danny brings to PCM his extensive experience with small business computers and applications software.)

the "TYPE" command. For now, though, our concerns are with COPY.

Think about this command:

```
COPY CON: A:TESTFILE
```

This would copy whatever is entered from the keyboard to "TESTFILE" on drive A:. If the file was not previously on the disk, it will be created. If it is on the disk, the contents will be erased in favor for the new data that is being entered.

When you use COPY this way, the cursor drops down one line and you can type anything you like. As you type, the data is written to the file on the disk. When you are finished, you type the "end-of-file" character CONTROL Z which tells the COPY command that it has "read" all the data from the keyboard.

Let's try a practical example of this function of COPY. With a backup of your MS-DOS master diskette in drive A:, type the following:

```
COPY CON: A:TESTFILE  ENTER
REAGAN                ENTER
CARTER                ENTER
FORD                  ENTER
NIXON                 ENTER
JOHNSON               ENTER
KENNEDY               ENTER
                  CONTROL Z
```

When you press CONTROL Z, the file is saved to the disk and control is returned to MS-DOS. (To get CONTROL Z, hold down CONTROL and strike 'Z'. Use the CONTROL key as you would SHIFT.

You've created "TESTFILE" on drive A:. If you don't believe it, type this command to display the contents of the file on the screen:

```
COPY A:TESTFILE CON:   ENTER
```

Aren't computers wonderful!

COPY is often used to create short files on a disk. The problem is that once you have the file on the disk, the only way that COPY could change it would be to retype the entire file. This is fine for short little files but for a larger one, you need a more powerful command.

Enter EDLIN

EDLIN is a line-oriented word processor. It is not actually a built-in MS-DOS command, but an entire program with commands of its own. In addition to the file creation ability found in COPY, EDLIN allows you to edit or add on to existing files.

Let's use EDLIN to create a file that we'll call "FILE2" (how imaginative). To start editing FILE2 with EDLIN, type this command:

```
EDLIN A:FILE2     ENTER
```

This tells EDLIN that we are working with "FILE2" on drive A:. If the file was not already in the directory (as in this case), it would be created, otherwise, it would be loaded into memory and we'd be allowed to edit it.

EDLIN will display the following:

```
New file
*—
```

The asterisk (*) is the EDLIN prompt. It tells you that EDLIN is waiting for your command. EDLIN has its own set of commands totally different from those used in MS-DOS. The first thing we want to do is to "insert" data into this newly created file. The command for insert is "I." Type:

```
I     ENTER
```

The screen should now look like this:

```
New file
*I
  1:*—
```

The "1:*" tells you that you are working on line one of the file.

Type these lines:

```
BANANA
  SPLIT            ENTER
APPLE PIE          ENTER
PEAR PUDDING       ENTER
CHERRY SODA        ENTER
PEACH MELBA        ENTER
GRAPE JUICE        ENTER
```

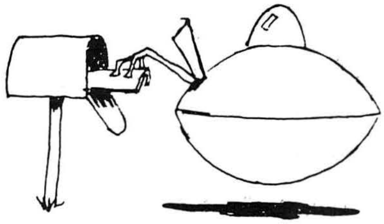
When EDLIN displays the next line number (7:*), press CONTROL C. This exits the insert mode and returns you to the asterisk prompt.

Now tell EDLIN to display the third line by typing:

```
3       ENTER
```

The screen will look something like this:

```
*3
  3:*PEAR PUDDING
  3:*—
```



You are may now make changes to this line or press ENTER to return to the EDLIN command prompt.

Let's change "PEAR PUDDING" to "PLUM PUDDING." Personally, the idea of plum pudding is no more appetizing to me than pear pudding, but just for example's sake. . .

Press the right arrow key [→] once. The letter 'P' should appear and the cursor will move one space to the right. This key tells EDLIN to copy one character from the "template" (the original line) to the new line.

Now type "LUM" to finish the letters of "plum." When you type a character, you overwrite the corresponding character in the original line.

We are now up to the space in the template. At this point, we could just press ENTER and the line would read "PLUM," or we could continue editing. We want to continue.

Press [→] three times and the line should now look like this:

```
3:*PLUM PU—
```

You've just copied three more characters from the template, the space and 'P' and 'U.' You could continue to use the right arrow key until the line reads "PLUM PUDDING," or you could use a special function to copy to the end of the template. Press the F3 function key.

The cursor will jump over and the screen will now look like this:

```
3:*PLUM PUDDING—
```

Press ENTER to store this updated line.

EDLIN also allows you to *insert* characters into a line template. Let's change "BANANA SPLIT" to "BANANA NUT SPLIT." We need to insert "NUT" and another space between "BANANA" and "SPLIT." "BANANA SPLIT" is in line one of the file, so we need to tell EDLIN to allow us to edit that line by typing:

```
1          ENTER
```

The screen will look like this:

```
*1
   1:*BANANA SPLIT
   1:*—
```

Press [→] five times to reveal the word, "BANANA." The screen will now look like this:

```
*1
   1:*BANANA SPLIT
   1:*BANANA—
```

Locate the key marked INSERT (above the numeric keypad on the right side of the keyboard) and press it. You have turned on the "insert mode." any characters you type now will be inserted into the template instead of replacing template charactes. When you press INSERT again, you turn off the insert mode.

Press the spacebar once and type the word, "NUT." The screen will now appear as this:

```
*1
   1:*BANANA SPLIT
   1:*BANANA NUT
```

A space and the word "SPLIT" is remaining in the template.

Turn off the insert mode by pressing INSERT. Now press [F3] to copy the remaining characters from the template into the edit line and you'll see this on your screen:

```
*1
   1:*BANANA SPLIT
   1:*BANANA NUT SPLIT—
```

Press ENTER to store the edited line.

Do we really want nuts on our split? Let's take them off and save the calories! Press '1' and ENTER to edit line one. Press the [→] key seven times to copy the word "BANANA" and a space to the edit line. The screen will look like this:

```
*1
   1:*BANANA NUT SPLIT
   1:*BANANA—
```

We now want to delete the next four characters ("NUT" and a space). This is quite simply done by pressing the DELETE key (located near INSERT) four times. Each time you press DELETE, a corresponding character is removed from the template.

Press [F3] copy the remaining characters from the template and your screen will look like this:

```
*1
   1:*BANANA NUT SPLIT
   1:*BANANA SPLIT—
```

Press ENTER to enter the edited line into our file.

By now, our screen is full of lines of editing and we cannot see the other lines of the file. This is no problem. One of the commands available to you in EDLIN is the 'P' (Page) command. 'P' allows you to list a series of lines to the screen. Type this:

```
P        ENTER
```

Since line one was the last line we worked on, it is the "current" line. 'P' will display the list we entered.

If you don't tell 'P' otherwise, it will list a page (a screen full) of lines starting with the line after the current line. If the current line is one (as is the case here), then it is the first line listed. Like most single-letter commands available in EDLIN, 'P' allows you to specify a beginning and ending line number for a range. The format for 'P' is:

```
[line 1][,line2]P
```

Where [line1] and [,line2] are optional line numbers. Try this:

```
1,3L        ENTER
```

This tells EDLIN to list lines one through three and the screen will look like this:

```
*1,3P
  1:BANANA NUT SPLIT
  2:APPLE PIE
  3:*PLUM PUDDING
*_
```

Notice the asterisk at the third line. The asterisk marks the current line. When you use 'P', the current line is changed to the last line displayed.

If you omit the second line number, 'P' assumes that you want to print to the end of the file or until the screen is full. Try this:

```
3P        ENTER
```

EDLIN will list the file starting with the third line. The current line is now line six.

If you omit the first line number, 'P' will assume that you want to begin with the line following the current line. The format to do this would be:

```
,nP
```

The 'n' would be a legitimate line number.The comma tells EDLIN that we are not specifying a starting line. The current line is now six (the last line), so it won't do much good to use this format of 'P' now.

Up to this point, everything we've done has been in the computer's memory. The idea is to get this stuff into our disk file. This happens when we exit EDLIN by pressing 'E' and ENTER. 'E' (end) saves the memory contents and returns control to MS-DOS. Go ahead and press 'E' to save our file and get out of EDLIN.

Another way of exiting EDLIN is 'Q' (quit). This command returns to MS-DOS without saving the changes that were made to the file.

There are many more commands available with EDLIN than we have room to discuss this month. Some of the goodies available let you move lines around, search for words in the file, globally replace words, and so many other nifty things. We'll take a look at those in September. Until then, get brave and do some exploring on your own. The "EDLIN" section of your MS-DOS manual should make more sense to you after what you've done today.

Telecommunicating With PoCo

Going online with this major news service puts you directly in touch with much of the data you need for informed financial decision making.

Dow Jones Means Business

By Randy Graham
PCM Contributing Editor

Okay, down off last month's soapbox. Let's resume our electronic odyssey.

For the past few months, we have been on a sightseeing tour of popular information services. This month, the bus stops at the third in the series: Dow Jones. Just to take a quick glance over our shoulder, we said that you probably met Dow Jones News Service at the Radio Shack store when you bought a modem cable for your PoCo. Included in the package was a free subscription to CompuServe and one to Dow Jones. We have already visited CompuServe and The Source (which we discovered elsewhere). On with the tour.

Your faithful guide must confess that this is where he starts mumbling and stuttering. I know less about DJ than any of the others, simply because I do not have much use for it. There are a great many people who use stock market type information all day in their work. It seems important for them to have the latest, most complete and reliable data. For many, many others, the stock market is a major spectator sport. They follow "the Dow" like some folks do the NBA, keep up with the performance of the companies in which their pension fund invests, and keep up-to-date on the "stars." I must confess that I am not among them. I do not need market or financial information at work, nor do I follow the market for fun. And so, I seldom logon to Dow Jones except for research purposes.

(Randy Graham is a rehabilitation counselor working with the handicapped. Personal computing is his hobby; telecommunications, one of his favorite activities. He has done freelance information retrieval and is an inveterate user of the major online systems.)

What you find is what you would expect: corporate information about publicly owned stocks, market quotes, histories of performance of stocks, etc. If you want it, this is an authoritative source. I feel that if you are a Dow Jones type, I am telling things you already know. The chance to keep up with the market may have been the main reason you acquired a PoCo.

What I can tell you is that if you have been building up some experience on bulletin boards, CompuServe and The Source, you will not have any trouble with DJNS. They provide very clear, complete manuals to help you get acquainted. When you logon, you will be greeted by an online newsletter, which is free, and then a menu. As on most services now, HELP is only a "slash command" away. You can also branch off to MCI Mail for which DJ does not charge connect time. MCI Mail charges are only generated when you send mail.

There are other databases on Dow Jones. There is a news service which is frequently updated, backed up by an online encyclopedia for background information. You can read reviews, plan airline schedules and order merchandise. All of these services have the business person in mind. When not buying and selling stocks, they travel and entertain, etc. The news stories cover all areas, but from a business perspective, like the Wall Street Journal, its hardcopy cousin.

Now I am going to say something almost negative, something I try to avoid in this series. As a mass marketing strategy, DJNS has been stressing its general appeal. If you are not a business or financial type, will you get much from Dow Jones? Not really. Most of the general services are available on other services. But to be fair, if you can join free and there is no minimum charge, and costs are modest, why not? You can always sample their databases, show off to your business friends and pick up tidbits to drop into the conversation at the neighborhood cookout. Personally, I get all the news I want as fast as I need it from the papers and news magazines.

Well, that didn't take long. I think we have time to stop by another information service I recently discovered. Everybody back on the bus!

Delphi

There is an information utility in the Boston area, known as Delphi, which has all the marks of having outgrown the bulletin board stage and is getting into the class with CompuServe and The Source. It is owned and operated by Boston Telecomputer. It has the usual bulletin boards, conferences, mail, games, library, etc. It has more services and features than a major bulletin board like the Rainbow Connection, but not as much as you will find on The Source or CompuServe. One of its cornerstones is its online Kussmaul Encyclopedia.

I found it interesting, but not necessary to my needs. I am sure that more familiarity will lead to discovery of unique features, "neat" programs not found elsewhere and new friends. One problem a PoCo user will have is the long main menu which will scroll right off into oblivion if you do not stop it. It is a very user-friendly system with a lot of HELP messages. I must confess that I had never heard of the Kussmaul Encyclopedia. I tried it out a couple of times, looking for information on current political candidates without success.

Access to Delphi is via Tymnet and charges are very modest: $6 per hour. Subscription is advertised at $20. For more information, call them at 1-800-544-4005 (say you heard about them in PCM).

Miscellaneous

Notes pile up. We must occasionally clean out the old RAM files to make room for new projects:

If you subscribe to an information service, you have found that they all publish magazines and newsletters to keep you up-to-date: CompuServe's *Online Today,* Dow Jones's *Dowline,* the Source's *Sourceworld*, etc.

Today's latest issue features an article by their Executive V.P. on "Unauthorized Access" — still a lively topic. I read your article, Mr. Minot; did you read mine? As I write this, the popular press has been excited again about someone breaking into a national credit file system. Please note that the problem seems to have been that a dishonest or careless employee who was entitled to the access code at work let it fall into the hands of unauthorized person(s) who put it up on a bulletin board. As usual, the security problem was not at the system end but at the user end. If you do not lock your door, the police will investigate the burglary, but they will not be very sympathetic! As I said last month — system security is adequate; end-user security is up to you.

If you like to read about telecommunications, *Link-Up Magazine* is supporting the field. For subscription information, write them at P.O. 26345, Minneapolis MN 55426 (mention PCM).

A recent birthday brought gifts of *The Computer Phone Book* by Mike Cane (New American Library, 1983) and *Omni's Online Database Directory* by the staff of *Omni Magazine*, Robert Weil, editor, et al. (McMillan, 1983). I find both useful references although bulletin boards and databases come and go so fast that printed materials tend to get out of date before they reach the market. It is hard enough to keep a magazine current!

Writing a column or series is very educational. It no sooner appears in print than people you never heard of call or write to tell you what you said wrong. Thanks to a CompuServe friend (whose name got lost in the pile of notes) for showing me that you can indeed send yourself Email on CompuServe. This is useful for practice and also to keep "carbons" in your file space.

I also said that you cannot "CHAT" privately with another subscriber on CompuServe as you can on The Source. Wrong. CIS's command is "TALK". On both systems you have to discover that they are online by recognizing ID numbers and getting their "JOB" numbers; i.e., which port they are working at the time.

Well, that's enough work for a hot summer day. Let's head for the beach!

PCM

A Few Good Bars (Cover Charge Only)

By Danny Humphress
PCM Technical Editor

Versions of *Lotus 123, Supercalc*^{3} and *Open Access,* those fancy spreadsheet/graphics packages, are finally available for the Model 2000. You may find, though, that you don't want to pay for the multitude of features that these fine packages offer. If all you want from your Model 2000 is a simple bar graph, it can be yours for just the cover price of this magazine and a little typing work.

This unadorned little program will produce a nice looking color bar graph on your Model 2000 display and optionally print it on a CGP-220 ink jet printer. Nothing fancy, mind you — just an easy-to-read bar graph.

There are three main functions of the program which may be selected from its menu. The first selection allows you to tell the program the title of your bar graph, an X axis title, and the number of bars in the graph. Selection two allows you to enter the actual figures and titles for the individual bars in the graph. The third selection displays your bar chart on the color monitor and optionally print it on a CGP-220 printer.

The program automatically adjusts the Y axis range for any value from zero to 100,000. For numbers larger than 100,000, divide by a factor of 10. If your highest value is 10 million, just enter 10. The width of the bars is automatically adjusted depending upon the number of bars in your graph. The maximum number of bars was made seven to match the number of colors available in addition to the white background.

You may change entered data at any time by simply selecting function 1 or 2 and re-entering the information.

The program will run without modification if you will not be using a CGP-220. If you want this feature, you'll need to change the first line to read as:

1000 SP=1

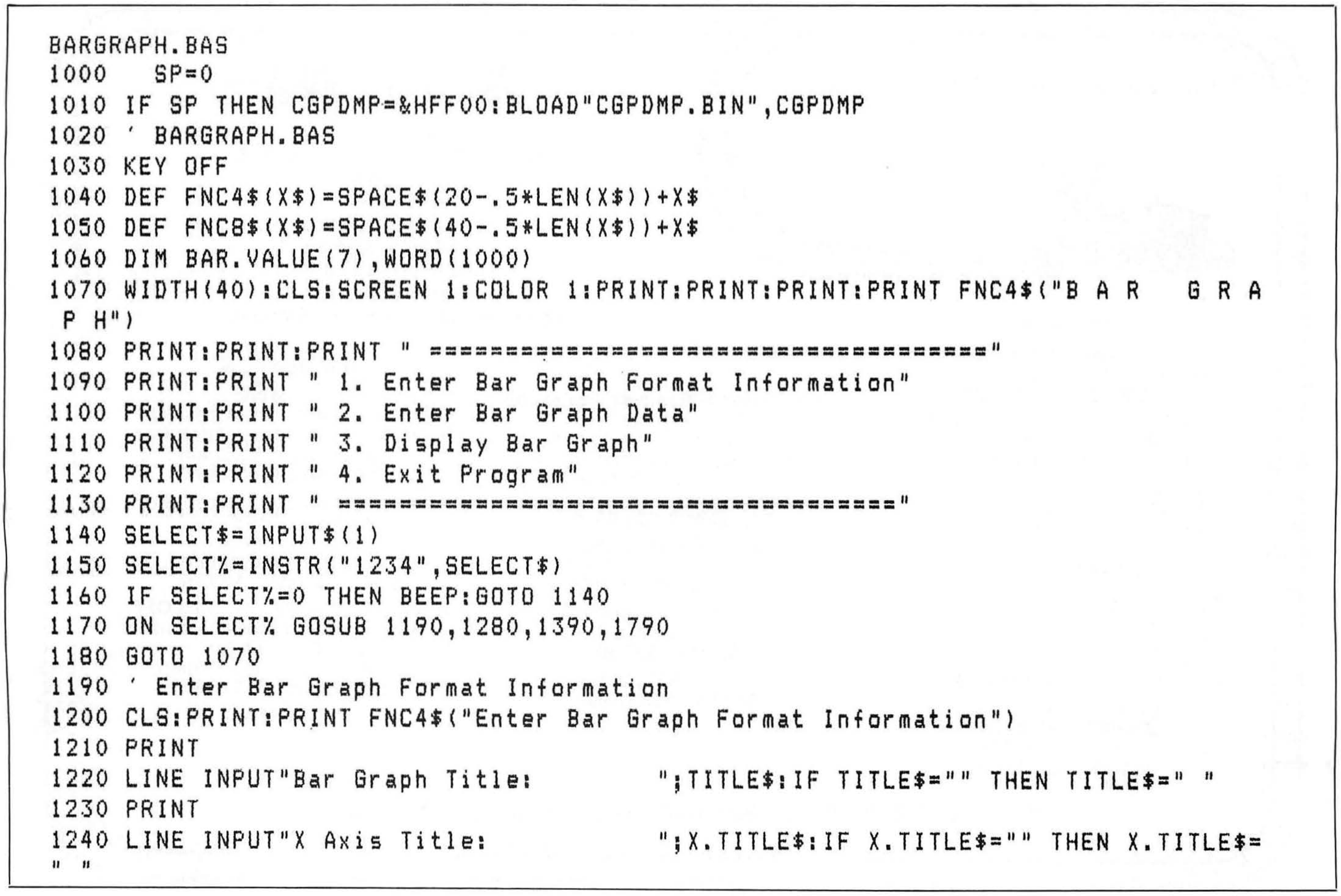
This enables the screen print option. Once this change has been made, you are given the option to print the screen as you display it.

You'll also need to make sure you have *CGPDMP.BIN,* a graphics dump program included on all Tandy 2000 MS-DOS master disks, on your defult drive. Enter BASIC using the following command line:

BASIC/M:&HFF00

This sets aside memory for the *CGPDMP.BIN* screen dump program.

The listing:



```
BARGRAPH.BAS
1000   SP=0
1010 IF SP THEN CGPDMP=&HFF00:BLOAD"CGPDMP.BIN",CGPDMP
1020 ' BARGRAPH.BAS
1030 KEY OFF
1040 DEF FNC4$(X$)=SPACE$(20-.5*LEN(X$))+X$
1050 DEF FNC8$(X$)=SPACE$(40-.5*LEN(X$))+X$
1060 DIM BAR.VALUE(7),WORD(1000)
1070 WIDTH(40):CLS:SCREEN 1:COLOR 1:PRINT:PRINT:PRINT:PRINT FNC4$("B A R   G R A
 P H")
1080 PRINT:PRINT:PRINT " ====================================="
1090 PRINT:PRINT " 1. Enter Bar Graph Format Information"
1100 PRINT:PRINT " 2. Enter Bar Graph Data"
1110 PRINT:PRINT " 3. Display Bar Graph"
1120 PRINT:PRINT " 4. Exit Program"
1130 PRINT:PRINT " ====================================="
1140 SELECT$=INPUT$(1)
1150 SELECT%=INSTR("1234",SELECT$)
1160 IF SELECT%=0 THEN BEEP:GOTO 1140
1170 ON SELECT% GOSUB 1190,1280,1390,1790
1180 GOTO 1070
1190 ' Enter Bar Graph Format Information
1200 CLS:PRINT:PRINT FNC4$("Enter Bar Graph Format Information")
1210 PRINT
1220 LINE INPUT"Bar Graph Title:          ";TITLE$:IF TITLE$="" THEN TITLE$=" "
1230 PRINT
1240 LINE INPUT"X Axis Title:             ";X.TITLE$:IF X.TITLE$="" THEN X.TITLE$=
" "
```

```
1250 PRINT
1260 LINE INPUT"Number of Bars (7 Max): ";BAR$:BAR%=VAL(BAR$):IF BAR%<2 OR BAR%>
7 THEN BEEP:GOTO 1260
1270 RETURN
1280 ' Enter Bar Graph Data
1290 CLS:PRINT:PRINT FNC4$("Enter Bar Graph Data")
1300 HIGH=0
1310 FOR I%=1 TO BAR%
1320    PRINT
1330    PRINT USING "     Bar # ";I%;
1340    LINE INPUT "Value: ";VALUE$:BAR.VALUE(I%)=VAL(VALUE$)
1350    LINE INPUT "           Title: ";BAR.TITLE$(I%):IF BAR.TITLE$(I%)="" THEN
 BAR.TITLE$(I%)=" "
1360    IF BAR.VALUE(I%)>HIGH  THEN HIGH =BAR.VALUE(I%)
1370 NEXT I%
1380 RETURN
1390 ' Display Bar Graph
1400 AN$="":IF NOT(SP) THEN 1430
1410 LOCATE 10,4:PRINT"Do you want to print this graph?";
1420 AN$=INPUT$(1):IF INSTR("YyNn",AN$)=0 THEN BEEP:GOTO 1420
1430 SCREEN 3:WIDTH(80):CLS
1440 PALETTE 0,0:PALETTE 1,4:PALETTE 2,2:PALETTE 3,6:PALETTE 4,1:PALETTE 5,3:PAL
ETTE 6,5:PALETTE 7,7
1450 LINE(0,0)-(639,399),7,BF
1460 LINE (50,19)-(620,351),0,B
1470 IF HIGH<=10 THEN TOP=10:INCR%=1:GOTO 1510
1480 FOR J=0 TO 100000! STEP 50
1490    IF HIGH<=J THEN TOP=J:INCR%=J/10:GOTO 1510
1500 NEXT J
1510 BAR.WIDTH=560/BAR%
1520 FOR I%=1 TO 9:LINE (50,33*I%+20)-(620,33*I%+20),0:NEXT I%
1530 FOR I%=1 TO BAR%
1540    Y%=350-(BAR.VALUE(I%)/TOP )*330
1550    X%=(I%-1)*BAR.WIDTH+50
1560    LINE (X%+10,350)-(X%+BAR.WIDTH,Y%),I%-1,BF
1570    LINE (X%+10,351)-(X%+BAR.WIDTH,Y%),0,B
1580 NEXT I%
1590 WORD$=TITLE$:GOSUB 1740:PUT(320-.5*LEN(WORD$)*8,2),WORD,PRESET
1600 FOR I%=1 TO BAR%
1610    X%=(I%-1)*BAR.WIDTH+55+.5*BAR.WIDTH
1620    WORD$=BAR.TITLE$(I%):GOSUB 1740:PUT(X%-.5*LEN(WORD$)*8,354),WORD,PRESET
1630 NEXT I%
1640 WORD$=X.TITLE$:GOSUB 1740:PUT(320-.5*LEN(WORD$)*8,375),WORD,PRESET
1650 FOR I%=10 TO 0 STEP -1
1660    Y%=33*I%+15
1670    LOCATE 1,1:PRINT USING "##### ";(10-I%)*INCR%
1680    WORD$="      ":GOSUB 1760
1690    PUT(0,Y%),WORD,PRESET
1700 NEXT I%
1710 IF SP THEN CALL CGPDMP(ER%)
1720 IF INKEY$="" THEN 1720
1730 RETURN
1740 '
1750 LOCATE 1,1:PRINT WORD$+"  "
1760 GET (0,1)-(8*LEN(WORD$),13),WORD
1770 LINE (0,0)-(8*LEN(WORD$)+16,16),7,BF
1780 RETURN
1790 ' Exit Program
1800 SCREEN 1:WIDTH (80):CLS:END
```

Your 2000 Can Help Access Those Applications

By Lawrence C. Falk

Last month we talked about how you can organize files in your Tandy 2000 so that you could have one application program in a single directory.

One of the advantages of this setup is that it enables you to keep things together and make individual directories manageable. This is particularly so when you have programs, such as *Multiplan*, which create files you will want to look at and use again.

However, there is a slight disadvantage, too. You have to *get to* the directories stored on either your hard or floppy disk. If you are "into" computers, this can be fun — typing mysterious commands to move yourself around to the place you want to be.

But, let's face it — most of the reason for having a computer in the first place is to make life easier for all of us. And, remembering a bunch of commands isn't a very easy thing to do. So, what we'll do here is show you how to let your 2000 do the work for you!

The example we will use will be with *Multiplan*, although you can use the same thing with almost any application. The reason for this is simply that we will be using the operating system, not the application programs. So, what we are going to do here will work with any application program. And, for that matter, you can make things more fancy if you want — just use a little imagination.

You may recall last month (PCM, July, Page 19) we created a directory in which to store *Multiplan* and called that directory "$". This was on the theory that the program deals primarily with dollars and cents, and it was something that was easy to remember. After we created the directory, we moved all the *Multiplan* "programs" into the "$" directory.

Whenever we wanted to run *Multiplan*, we just typed **CD \$** and we "moved" from the main directory to the one which held *Multiplan*. Then, we simply ran the program as per the instructions. Once we were through, we quit *Multiplan*, typed the command **CD ** and were back in the main (or root) directory of the disk.

Now, what we want to do is eliminate all that remembering and typing. We can do that by using a nice little program in the MS-DOS system called **EDLIN.**

EDLIN is a simple line editor. To run it, go to the root directory and type its name, along with the name of the program or procedure that you want to use it to create.

In this case, we selected "$" to be the directory for *Multiplan* because it was easy to remember. No reason to change things, so let's just call the procedure we will use to get *to* it "$", too.

(You probably know that you cannot use two file names that are the same. However, when you are dealing with different *directories* you can use the same file name in each of any number of directories you are using. MS-DOS treats each directory as a different disk in some ways and that is the reason this is possible. As you'll see, this can be very useful.)

OK, let's go. You have a directory

called "$" with *Multiplan* programs in it. You are in the root directory. Just type:

```
EDLIN $.BAT
```

What you have done is started to run the **EDLIN** program and given the file you are creating a name, in this case "$". Your 2000 prints the words "New file" on the screen and an asterisk (*). The asterisk is your prompt to go ahead and do something.

Type the following **EDLIN** command:

```
I
```

This puts you in the Insert mode, and tells the **EDLIN** program you are ready to begin inserting lines.

EDLIN responds by typing the line number, a colon and another asterisk, like this:

```
1:*
```

Now you are going to type in what you want this procedure file to do. Immediately following the asterisk, you type:

```
chdir \$
mp
```

To get out of the Insert mode you hold the CONTROL key down and type a 'Z,' followed by ENTER. Then, type an 'E' and ENTER to get you out of **EDLIN** and back to the "command level."

That's all there is to this part of it. What you have done is to create a procedure file which changes the directory from the root to the "$" directory and then runs *Multiplan*. The name of the file procedure file is *$.BAT*, but you can run it from the main directory simply by typing in "$" and ENTER.

Try it. What you will see is that your new file changes your directory from the root to the "$" directory and then loads in *Multiplan*. All with one keystroke!

Well, truth be it known, when you have finished with *Multiplan* it is always good to save files, that is, make a backup. Because we use a hard disk, we have two methods of saving, and the most simple is just to make a second copy of the same file on the hard disk. We do this because it is faster and because we can always insert a floppy and make another backup. We call this a secondary backup and do it only once in a while on the theory we may ruin a single file, but not two of them.

So, what we need to do is create a procedure file not only to make some backups, but also to get us back to the main directory (after all, once done with *Multiplan* why hang around in its directory?).

Just for illustration purposes, let us suppose that we have two files that we use with *Multiplan* every day, one called **SUBS** and one called **MONTHS**. Each day we turn on our 2000, type "$" after we enter the date and time, and then go into the two files, updating information based on the number of subscriptions and money we get every day. Once finished, we give *Multiplan* its **QUIT** command and we are back, sitting in the "$" directory.

> *"What you will see is that your new file changes your directory from the root to the '$' directory and then loads in* Multiplan. *All with one keystroke!"*

So, now, let's just create a file that will get us back to the root directory and save copies of these files at the same time.

Suppose we want to call the backup files **B-SUBS** and **B-MONTH**, with the prefix **B-** standing for backup. Type in the command to change back to the root directory (you have to do this because **EDLIN** is only in the root directory) by using the now familiar command **CHDIR \**.

Okay. Back to the root directory. Since we are going to use this new procedure file primarily to be safe and make a copy of existing files, let's name it **SAFETY**. So, to get into **EDLIN** we type in:

```
EDLIN SAFETY.BAT
```

We get the "New file" message and the asterisk. So, type in the command I (for insert) and note that **EDLIN** supplies the same numeral 1, the colon and an asterisk. Now, we are ready to create the new file, so type in the new procedure this way:

```
COPY SUBS B-SUBS
COPY MONTH B-MONTH
DIR *.
CHDIR \
```

Now, type a CONTROL Z to get out of the insert mode, type an 'E' to get out of **EDLIN** and you have a new procedure file named *SAFETY.BAT*.

But, this procedure file is in the root directory (where we created it) and it needs to be in the "$" file. Simple matter to handle this here, just type the following two commands:

```
COPY SAFETY.BAT \$\
   SAFETY.BAT
DEL SAFETY.BAT
```

What you have done is copy the procedure file from the root to the "$" directory and then deleted the same file from the root directory so the root directory does not have a lot of additional clutter in it.

Your new procedure file will do four things; it will:

Copy **MONTH** into a file called **B-MONTH**
Copy **SUBS** into a file called **B-SUBS**
Print out a directory on the screen so you can see the copies were, indeed, made
Returns you to the root directory.

When in the "$" directory, all you have to do is type the procedure file name **SAFETY**. All the rest of the work will be done for you.

Now that you have these two procedure files, all you need to do to run *Multiplan* on a daily basis is turn on your 2000, enter the date and time, and type in "$" ENTER. You will be sent to the proper directory and *Multiplan* will start up. You do whatever work you wish to do in *Multiplan*, **Q**uit the program and type in the word "SAFETY" and ENTER. The 2000 will automatically make a backup of both files, print a directory on the screen and return you to the root directory, where you can do something else.

One last point. Remember that we said that you could have file names which are the same so long as they are on different directories? Well, this means you could use the name "SAFETY" as a backup procedure file on every application program you have — just as long as each application program is in a directory all its own. It makes things somewhat easier to remember.

As you get more familiar with procedure files such as these, you will, no doubt, come up with some additional things you want them to do. It is easy to do so, once you have the basics down, and this article should lead you through that easily.

The PoCo-CoCo Connection

By D.E. Mitchell

My first home computer was a TRS-80 Color Computer, which over an approximately two-year period grew from a 16K machine with a cassette recorder and an LPVII printer to a 64K, two-drive system with a DMP-200 printer and a modem, a case with no screws left, all sorts of funny wires running around inside it, and two shiny black thumb prints on its base.

Then, Radio Shack announced their "notebook" computer, the Model 100. I played with one a little while, yawned a bit, allowed that it was "nice," and promptly forgot about it and forged on with my Color Computer. Then technology caught up, and the buzzwords of the day were modems, data communications and multi-computer linkages. A little bell went off in my head, reminding me that a lot of those words seemed to have been in the Model 100 manual that had been so casually laid aside. So, back to the manual, and sure enough . . . the machine was designed for the urges that I was attempting to satisfy with the Color Computer. So down to my friendly Radio Shack salesman and the order was in for a 24K Model 100.

Once I got the unit and worked with it for a while, I was alternately impressed and depressed with its capabilities. True, it did, and did very well, a lot of things that I was used to doing. It did have excellent built-in software, it did communicate easily, it was remarkably easy to learn and use. However, it didn't have disk storage (how I've learned to hate cassettes), it didn't have a good word processor, and there was no spreadsheet of the capabilities I was used to. Now I knew that either Radio Shack or a third-party vendor would soon provide some of these things (and some are already here as I write this), but I was eager to use the machine that minute, and really didn't want to put a whole lot more money into it. Another little bell went off — I realized that what the Model 100 didn't do, the Color Computer did, and vice-versa. Introduce a little symbiosis and Voila! — the idea of the PoCo-CoCo connection was born. Why not connect the two systems so that each did the things that they could do the best.

Like any good planning session, the first step in setting up something like this is to define the goals and objectives. What did I really want to do? I came up with the following: I wanted to communicate with bulletin boards and tap into all that fine camaraderie, knowledge, and shared software. And, I wanted to pass data back and forth between the Model 100 and the Color Computer in order to use the CoCo's disk drives for storage of PoCo programs and text files, and to be able to use the Color Computer word processor for polishing my Model 100 scribblings.

Very gradually, with lots of help from friends with varying experience and a few more minor purchases, these goals have been realized. For those of you who might have the same basic setup, or might want to take advantage of the current ridiculously low prices being asked for Color Computers, or might even want to transfer the techniques to

(Dennis Mitchell is head of the business software group of Color Software Services, a division of Brantex, Inc.)

set up another computer as a base station for the Model 100, I will describe how the goals were met. I will be using the specific names and parameters of the software and hardware that I have used; however, many equally fine pieces of both should be able to be substituted to do the job as well, or better. It is just easier for me to describe them using terms that I have grown familiar with.

Communicate With Bulletin Boards

I had already been doing this with the Color Computer before I got the Model 100, using Eigen Systems *COLORCOM/E*, one of the many communications software packages available. However, the auto-find-and-dial capability of the Model 100 sure is a lot smoother, but the buffer of the 64K Color Computer is bigger, so I find that I still alternate between which one to use, depending on what I am looking for. If it involves a lot of downloading, the Color Computer does it. If it is just board-hopping or looking for one or two specific things, the vote goes to the Model 100.

> *"Like any good planning session, the first step in setting up something like this is to define the goals and objectives. What did I really want to do?"*

In my *ADRS.DO* file, I have preceded each bulletin board with the code "BBS." I can then use the find feature of *TELCOM* to scan through them until I find one that I want to call with the CALL function key.

Transferring Programs And Text Files Between The Model 100 And The Color Computer

This didn't seem like it would be too hard to do, since no phone connections were involved, only a direct link by cable between the two computers. Radio Shack had even anticipated this requirement, offering for the mere sum of $19.95 a cable to join the two (Cat. #26-3014). I hooked this cable to the RS-232 port of the Model 100 and the serial I/O port of the Color Computer, powered up *COLORCOM/E* in the CoCo, set the parameters on both machines to match (more on this later), pushed a couple of buttons, and . . . nothing happened.

After a couple of days of experimenting with every conceivable parameter in both machines, it was back to Radio Shack. It seems that they forgot to mention that I needed a null modem adaptor ($29.95, Cat. #26-1496).

"What's a null modem adaptor?" It's not too complex. When you hook the Color Computer and the Model 100 together via the 26-3014 cable, both are trying to send and receive signals over the same lines. All a null modem adaptor does in this case is cross them over in mid-stream. If you don't mind ruining this cable for other uses, simply unsolder pins 2 and 4 on the four-pin connection end or pins 2 and 3 on the 25-pin end, switch them around and resolder, connect it up and it should work. They did, I did, and it did.

The only thing remaining was to adjust the parameters on both ends for compatible data transfer. The fastest rate that *COLORCOM/E* can handle is 1200 Baud, with the following settings:

COLORCOM/E
Baud = 1200
Parity = Even
Bits = 7
CR = Normal
LF = Insert
CTL = Pass
Duplex = Full

(With the exception of the 1200 Baud and insert LF option, all of the settings are default values. You may just hit ENTER in response to them.)

Model 100 STAT = 57E1E
5 = 1200 Baud
7 = 7 bit word
E = Even parity
1 = 1 stop bit
E = XON/XOFF enabled

To link the two computers, simply bring *COLORCOM/E* up in the CoCo, set the parameters (Ct 1-5 and L from the menu), and hit the Space Bar to get a blank screen with the cursor in the upper left corner. Put the Model 100 in *TELCOM* mode, STAT as 57E1E, and hit function key 4 (TERM). Type a bit, and you will see that the two computers are "talking." Use the upload option of the Model 100 to pass a file to the Color Computer, catch it in the buffer, and write it to disk (CTL-5 and option "W"). Use the file transmit feature in *COLORCOM/E* to send a file to the Model 100, and catch it in the 100's RAM with the download feature. (See the *COLORCOM/E* and Model 100 manuals for more detailed instructions on these features).

Once the files have been saved to the CoCo's disk from *COLORCOM/E*, they are an ASCII text file, and may be accessed by an ASCII-based word processor or text editor. In my case, since I use Cognitec's *Telewriter-64* with the ASCII file feature for my editor, I write the file to disk with a filetype of "DAT" (which signifies an ASCII data file to *Telewriter*), and since I have a communications systems disk in drive 0 and a data disk in drive 1, I usually add a ":1" to the file when saving it. For example, to write a received file to a Color Computer disk, choose *COLORCOM/E* option "W," answer "NO" to the "Use capture characters" question, and name the file NAME.DAT:1. Later, when using *Telewriter-64* to review and clean up the file, you may call it up as file NAME:1, the filetype DAT will be assumed.

One thing to remember when transferring a file from the Model 100 to the Color Computer is that for some strange reason the Model 100 does not seem to transmit a line feed in the upload mode, which is why I chose the option of inserting line feeds in *COLORCOM/E*. When sending a text file and later loading it into *Telewriter-64*, this will all be adjusted by the word processor anyway. However, when you send a BASIC program (which must first be converted to a *.DO* file in the Model 100 by doing an ASCII save), any line feeds inserted in a line can play havoc with your program. The best way I have found around this is to specify the widest possible line width (132 characters) during the upload. Then load the file into *Telewriter-64* with the characters-per-line set at zero (0) so that it does not insert any more line feeds, check to see that all lines begin with a BASIC line number, delete any unnecessary CRs or LFs that may have broken up a BASIC line, and resave the file.

If you can then load it as a BASIC program without any "DS" (Direct Statement) Errors, you have done very well, indeed!

Radio Shack's TRS-80 Offers You Disk St

The Revolution Continues

Our Model 100 Portable Computer's built-in software and easy portability started a revolution in the way America computes. Professionals of all kinds—managers, reporters, ambulance drivers—have found the Model 100 a valuable tool they can use anytime, anywhere. Now, with the TRS-80 Model 100 Disk/Video Interface, you can turn your Model 100 into a versatile disk-based home or office system, too!

Greater Storage Capacity and Faster Access to Data

The Model 100 Disk/Video Interface features a built-in single-sided, double-density 184K disk drive to let you create, store and retrieve files quickly and easily. You'll find it much faster and more reliable than cassette storage. When you're ready to hit the road again, just load the files you need into your Model 100 and take off.

```
Files
FNEXT  .BA*
Ok

lfiles
SYSTEM VER 01.00.00
FORMAT   1       BACKUP       2
BACKUP.SMG 2     DSKMOD       1
153 K AVAILABLE
Ok

load"fnext
Ok

list
5 CLS
10 PRINT "FOR-NEXT LOOP EXAMPLE"
20 FOR I = 1 TO 10
30   FOR J = I TO 10
40   PRINT I,J
50   NEXT J
60 NEXT I
Ok
```

® Model 100 Computer torage and Video!

Attaches Easily to Any Television

The Disk/Video Interface also lets you enjoy a big 40-character by 25-line screen display on any TV. The larger screen is perfect for writing notes and reports or developing your own programs in BASIC. When accessing data by phone, you can read a full 25 lines of information without scrolling. And you can utilize all of Model 100's graphics characters.

No Computer Wizardry Required

It couldn't be easier to use. Just plug the Disk/Video Interface into an AC outlet, connect your TV and a Model 100 with a minimum of 16K—all cables are included. One simple command switches all the output to the video screen. There's no new operating system to learn, either—the command structure is exactly the same.

Easily Expandable

You can add a second 184K disk drive (#26-3807, $239.95*) to the Disk/Video Interface for even more versatility. Best of all, the Disk/Video Interface doesn't use any of Model 100's standard connectors, so you can add a printer, bar code reader, RS-232C communications device or cassette recorder. You can make your Model 100 a high-performance desktop system.

TRS-80 Model 100 Disk/Video Interface

799^{00} 26-3806

Shown with optional second drive

Get Down to Business With an 80-Column Display

A Compact Office System

The Disk/Video Interface can turn your Model 100 into a professional desktop system for the office, too. With any standard video monitor, you get a large 80-character by 25-line display that's equal to other desktop computers. You'll also get the rapid access to files and storage reliability that are so necessary in today's business world.

Perfect "Second System"

A disk-based Model 100 system is an economical alternative to a second office computer. And you'll still have the advantage of portability when you need it.

AS LOW AS $45 PER MONTH

Come In Today

The TRS-80 Disk/Video Interface, along with our complete line of Model 100 accessories, is available coast to coast.

Available at over 1100 Radio Shack Computer Centers and at participating Radio Shack stores and dealers

Radio Shack® COMPUTER CENTERS

A DIVISION OF TANDY CORPORATION

Send me a free TRS-80 Computer Catalog.

Mail To: Radio Shack, Dept. 85-A-13
300 One Tandy Center, Fort Worth, TX 76102

NAME ____________________

ADDRESS ____________________

CITY ____________________

STATE ____________ ZIP ________

TELEPHONE ____________________

*Installation not included. Prices apply at participating Radio Shack stores and dealers. TRS-80 Model 100, Disk/Video Interface and TV sold separately. CitiLine is a service mark of Citicorp.

3-D FOR YOUR LCD

Program by
Jon P. Klages

We have all seen 3-D graphics on CRT's but here is a little program that puts 3-D on the Model 100's LCD. And it only uses 612 bytes of RAM to do it.

The 'X' loop draws the function. The 'T' loop establishes the 3-D effect by plotting the 'X' function from a different angle each run.

Since the function is symmetrical, being drawn around a common center, it only has to be drawn through 180 degrees of rotation. Further, the function and its compliment are drawn simultaneously, therefore it is only required to go through 90 degrees which cuts generation time in half. It takes about three minutes 43 seconds to complete "SINC," three minutes 16 seconds to complete the "COSIN" and five minutes three seconds to complete the "HAT."

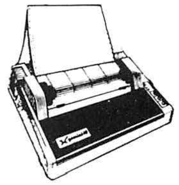
The formulas used in this program are, basically, X'=XCOS0 and Y'=Y+XSIN0 where X and Y are individual points on the actual graph plan and X' and Y' are coordinates on the view plane. Theta (0) is the angle between these two planes.

— **K.W. Klages**

PCM BAR CODED LISTING

The listing:

```
10 'GRAF3D.BA, AUTHOR JON P. KLAGES, (61
2 BYTES)
20 CLS:PRINT:PRINT:PRINT "1=SINC 2=COS 3
=HAT"
30 PRINT:PRINT "WHAT FUNCTION ?"
40 A$=INKEY$: IF A$=""THEN 40
50 IF VAL(A$)=1 THEN GND=35: GOTO 90
60 IF VAL(A$)=2 THEN GND=25: GOTO 90
70 IF VAL(A$)=3 THEN GND=25: GOTO 90
80 GOTO 20
90 PI=3.14159: P=239/2: Z=.2:CLS
100 FOR T=0 TO PI/2 STEP PI/31
110 G=COS(T): C=-P*G+P
120 H=SIN(T): D=GND+P*Z*H
130 FOR X=-P TO P STEP 5
140 IFVAL(A$)=1 THEN Y=4*PI*X/P: Y=20 *
SIN(Y)/Y
150 IFVAL(A$)=2 THEN Y=20 * COS(1.5*PI*X
/P)
160 IFVAL(A$)=3 THEN Y=1.5 * PI * X/P: Y
=18 * (COS(Y) -.4*COS(3*Y))
170 A=X*G+P
180 B=GND-Y-Z*X*H
190 IF T=0 AND X>1 THEN GOTO 230
200 LINE (C,D)-(A,B),1
210 LINE (239-C,D)-(239-A,B),1:C=A:D=B
220 NEXT X
230 NEXT T
```



Gemini 10X Printer (does not include cable)	**$299.95**
Mini-Modem (includes cable)	**$79.95**
C-10 Cassette Tapes	**$.49 each**
5¼″ Diskettes	**$1.99 each**

Rainbow Connection BBS

BBS #1 (212) 441-3755
2 (212) 441-3766
3 (212) 441-5719
4 (212) 441-5907
5 (408) 984-7937

Model 100 download available on these lines

RAINBOWfest
see us at *PRINCETON*

WEST DIVISION
Spectrum Projects
PO Box 9866
San Jose, CA 95157–0866

Add $3.00 S/H
NY Res Add Tax

EAST DIVISION
Spectrum Projects
PO Box 21272
Woodhaven, NY 11421

Part III—Files Of A Different Sort

By Danny Humphress
PCM Technical Editor

There is some exciting news on the dBASE front. Ashton-Tate, developers of dBASE II, have begun deliveries of its new enhanced version of dBASE, dBASE III. dBASE III adds several new features to our familiar dBASE II including an "Assist" feature that allows you to breeze through the program with the menus and help screens. Other major features include a new maximum database size of two billion records (v.s. dBASE II's 65535) and the ability to work with ten database files at a time (dBASE II allowed two). The disappointing news for us Tandy 2000 users is that the current version will work only on the IBM PC, PC/XT, PC*jr* and 100 percent compatibles. We expect a Tandy 2000 version soon.

dBASE II remains alive and vital. It will continue as the "standard" for 8-bit CP/M computers and an important part of Ashton-Tate's software line for 16-bit MS-DOS computers.

This column will continue to focus on dBASE II. Perhaps when dBASE III is readily available for the Tandy 2000, we'll incorporate it into dBASE Tutor. But for now, it's on to the miracles of dBASE II . . .

Over the past two months we've created a small database, added records to it, listed it, located records, and edited records. This month we're going to sort our list and learn why we don't *have to* sort. We'll also learn a quick way to find records without having dBASE search through the entire list.

We'll be using the database called "Mail" that we created in the first part of dBASE Tutor. If you're just joining this series (or you lost your disk from two months ago), Figure 1 has the structure and data for the file.

It is usually desirable to have information in a database in some order. In a short list such as ours, it is not so important, but imagine having to look through 1,000 items in an inventory file listing to find a single item. You might as well throw out the computer and go back to the card file system. Well, that's not a problem with dBASE II.

dBASE II provides a way to make a copy of a database file sorted in any manner you like. The command to do this wondrous thing is SORT. SORT takes the current database file in use and copies it to a new sorted file.

Enter these commands:

```
USE MAIL                  ENTER
SORT ON NAME TO MAIL2     ENTER
```

The first line, of course, tells dBASE II to use the "Mail" database file. The second line sorts the current file by the "name" field and creates a new database called "Mail2" to contain the sorted list. The current database file has not been changed at all. See for yourself — list the names by typing:

```
LIST NAME                    ENTER
```

The list you'll get will be in the order the names were entered — not in alphabetical order. The sorted list is in the new database called "Mail2." Tell dBASE II to use this file by entering this:

```
USE MAIL2                    ENTER
```

dBASE II is now using the sorted file that SORT created. Now try the LIST command:

```
LIST NAME                    ENTER
```

The list will be in alphabetical order by the "name" field. See Figure 2.

Now try this:

```
USE MAIL                     ENTER
SORT ON CITY TO MAIL2        ENTER
```

The "Mail2" file will now be sorted by the "city" field. Let's take a look.

```
USE MAIL2                    ENTER
LIST NAME,CITY               ENTER
```

We told dBASE to use the "Mail2" file and list the names and cities of all the records. The records should now be in alphabetical order by city. Isn't that easy?

Remember, SORT does not sort the file you're currently using, but rather it makes another copy of the file in sorted order. If the file you specify with the SORT command already exists, it will be erased and overwritten with the new information. It is very important that you are absolutely sure of what you're doing when you use SORT. It is very easy to forget, as in the above examples, that you already have a "Mail2" file that you don't want erased. Once you press ENTER, it's too late.

You Don't Need To Sort!

That's right, you don't need to sort. SORT is a nice function to have, but there are better ways of getting data in order. There are many disadvantages of SORT — all of which are overcome with another more powerful command.

To begin, SORT is time consuming. If you make any changes that would affect the way the data is sorted such as adding, deleting, or editing records, you have to sort the entire file again.

Figure 1

FIELD NAME	TYPE	LENGTH	DECIMAL
NAME	C	35	
STREET	C	35	
CITY	C	20	
STATE	C	2	
ZIP:CODE	C	10	
TELEPHONE	C	13	
YTD:SALES	N	10	2

1
PCM
9529 U.S. Highway 42
Prospect, KY 40059
(502) 228-4492
5100.50

2
Radio Shack
300 One Tandy Center
Fort Worth, TX 76102
(no telephone number)
123456.78

3
Portable Computer
Support Group
11035 Harry Hines Blvd.
Dallas, TX 75229
(214) 351-0564
1000.00

4
Computer Plus
480 King Street
Littleton, MA 01460
2000.00

5
B.T. Enterprises
10 Carlough Road
Bohemia, NY 11716-2996

6
Chattanooga Choo Choo
P.O. Box 15892
Chattanooga, TN 37415
4000.00

7
Dr. Preble's Programs
6540 Outer Loop
Louisville, KY 40228
5000.00

8
Computer Solutions Co.
901 Embassy Square Blvd.
Louisville, KY 40299-1814
(502) 491-6122
6000.00

9
Prickly-Pear Software
9234 E. 30th Street
Tucson, AZ 85710
(606) 886-1505
7000.00

10
Skyline Marketing Corp.
4510 W. Irving Park Road
Chicago, IL 60641
(312) 286-0762
8000.00

11
Purple Computing
2068 Ventura Blvd.
Camarillo, CA 93010
(805) 987-4788
9000.00

12
Spectrum Projects
4285 Payne Avenue #9866
San Jose, CA 95117
(212) 441-2807
10000.00

13
Dennison
82 Calvary Street
Waltham, MA 02154
(800) 343-8413
11000.00

14
Traveling Software, Inc.
11050 Fifth Avenue NE
Seattle, WA 98125
(800) 343-8080
12000.00

SORT wastes disk space. You have to sort to a different file. Of course, you could always erase the original file, but not until the new sorted file has been created. At one point, you must have two copies of the file on the disk.

One major problem is SORT's inability to have the data sorted in more than one order. In our "Mail" file, we may want the data in name order for reports, but also in ZIP code order for mass mailing. The only way to do this would be to re-sort the file each time you wanted to access it in a different order.

Another drawback is that, with a large file, we still have to search through

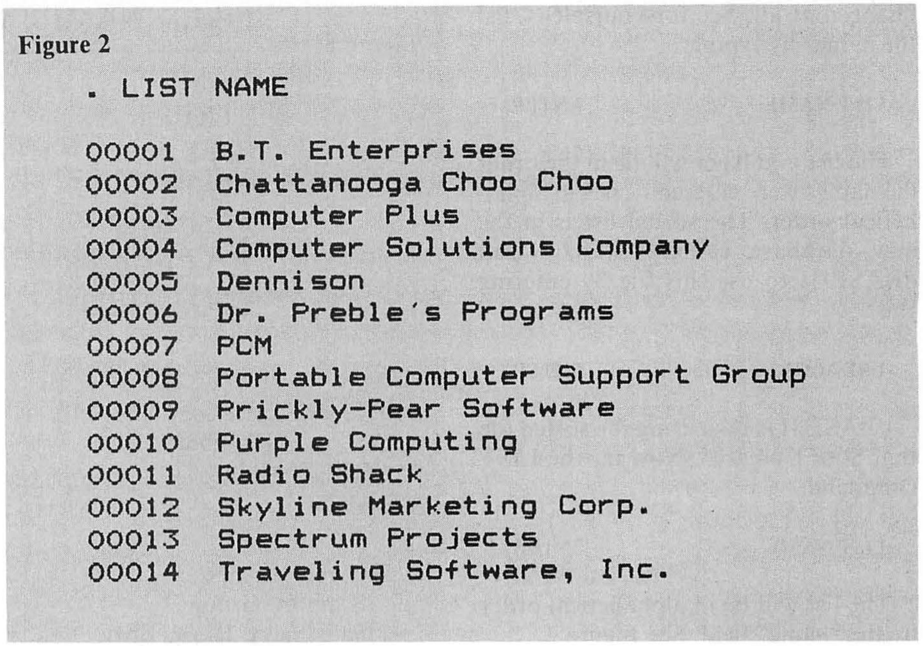


Figure 2

```
. LIST NAME

00001  B.T. Enterprises
00002  Chattanooga Choo Choo
00003  Computer Plus
00004  Computer Solutions Company
00005  Dennison
00006  Dr. Preble's Programs
00007  PCM
00008  Portable Computer Support Group
00009  Prickly-Pear Software
00010  Purple Computing
00011  Radio Shack
00012  Skyline Marketing Corp.
00013  Spectrum Projects
00014  Traveling Software, Inc.
```

the entire file with LOCATE when we're looking for a single record. Even if the file is sorted, dBASE starts with the first record and continues until it finds a matching record with LOCATE — very, very slowly.

There is an answer to all these problems — INDEX. The INDEX command creates an "index" to the database file. Much like the index at the end of a book allows you to find page numbers, INDEX helps dBASE find record numbers. The datafile is not physically sorted, but another "index" file is created that "points" to records in the database. Try this:

```
USE MAIL                          ENTER
INDEX ON NAME TO MAILNAME ENTER
```

A file called "MAILNAME.NDX" is created on the disk that has information about the "logical order" of the records in the "Mail" database. Now list the names in the file:

```
LIST NAME                    ENTER
```

Notice that the records appear in "name" order. But are they really? Look at the record numbers (Figure 3). The records were only displayed in order, they are actually still in the record number order. To turn off the index, type:

```
USE MAIL                     ENTER
```

The index is still on the disk, but as far as dBASE II knows, we're using a different database file, so it turned off the index. List the names:

```
LIST NAME                    ENTER
```

The names are still in the record number order. We don't need to go through the indexing again to get them back to name order, we just need to turn on the index. Type:

```
SET INDEX TO MAILNAME   ENTER
LIST NAME               ENTER
```

The first line told dBASE to use the "Mailname" index for this file. We can also tell dBASE which index to use as we tell it which database file to use by adapting a different form of the USE command:

```
USE MAIL INDEX MAILNAME    ENTER
```

This is a short way of telling dBASE to use the "Mail" file with the "Mailname" index. It does exactly the same thing as the two-liner above.

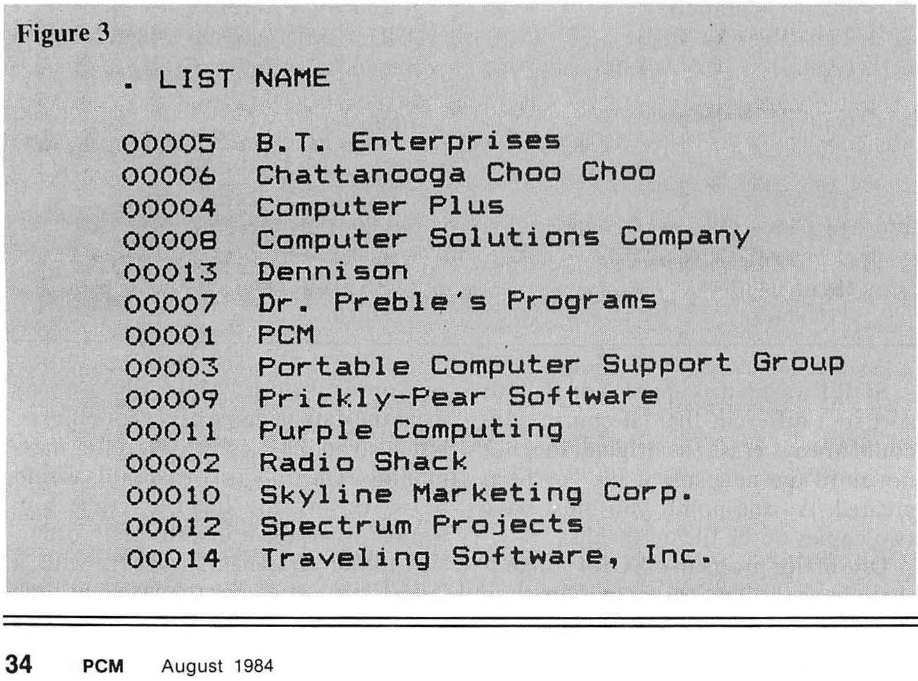


Figure 3

```
. LIST NAME

00005  B.T. Enterprises
00006  Chattanooga Choo Choo
00004  Computer Plus
00008  Computer Solutions Company
00013  Dennison
00007  Dr. Preble's Programs
00001  PCM
00003  Portable Computer Support Group
00009  Prickly-Pear Software
00011  Purple Computing
00002  Radio Shack
00010  Skyline Marketing Corp.
00012  Spectrum Projects
00014  Traveling Software, Inc.
```

Last month, we used LOCATE to find a record in the database. If the file in use is indexed on the field that you're looking for, such as "name," there is a much more efficient way. FIND uses the index to rapidly locate a particular record. No matter how large the database, FIND will locate a record in a matter of a few seconds or less. The only drawback is that you can only use it for the field that the database is indexed on. Try this:

```
FIND "Radio Shack"         ENTER
DISPLAY                    ENTER
```

The record with "Radio Shack" as the name was found very rapidly using the index. DISPLAY put the information for the record on the screen.

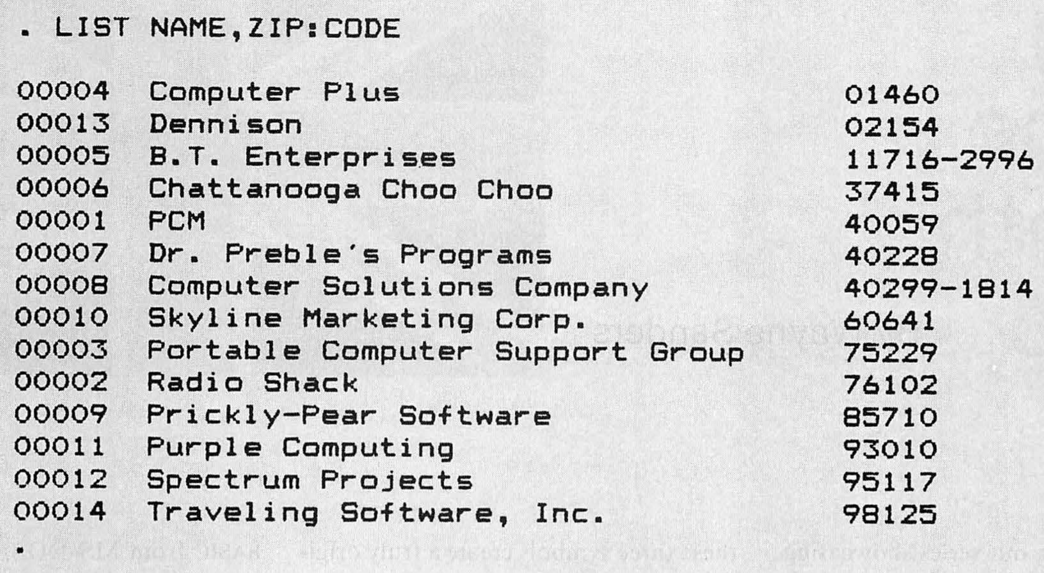
This is one very important feature of dBASE. With just a single command, you can quickly locate any record from thousands if the database is indexed accordingly.

Try this:

```
FIND "Sears"               ENTER
```

Since "Sears" is not in our database file, FIND reports a "NO FIND"

Figure 4



```
. LIST NAME,ZIP:CODE

00004  Computer Plus                          01460
00013  Dennison                               02154
00005  B.T. Enterprises                       11716-2996
00006  Chattanooga Choo Choo                  37415
00001  PCM                                    40059
00007  Dr. Preble's Programs                  40228
00008  Computer Solutions Company             40299-1814
00010  Skyline Marketing Corp.                60641
00003  Portable Computer Support Group        75229
00002  Radio Shack                            76102
00009  Prickly-Pear Software                  85710
00011  Purple Computing                       93010
00012  Spectrum Projects                      95117
00014  Traveling Software, Inc.               98125
.
```

message.

Like LOCATE, FIND will allow you to specify just a few characters to locate a record. For instance, "Chattanooga Choo Choo" is the only record in the file that begins with "Chat," so we could also do this:

```
FIND "Chat"                ENTER
```

dBASE will find the first (or only) record that begins with "Chat." This makes it very convenient to find a long name without having to type the entire name.

Another major advantage of INDEX over SORT is that you don't have to continually re-sort the file any time you add, delete, or edit a record. As long as you have the index turned on, it is updated automatically any time you fiddle with the data.

Try this:

```
FIND "Purple Computing"    ENTER
EDIT #                     ENTER
```

Change the name to "Yellow Computing" while in the edit mode. Now type:

```
LIST NAME                  ENTER
```

The new "Yellow Computing" has moved to a new position in our indexed list! Neat, huh! Type:

```
EDIT #                     ENTER
```

Change the name to "Purple Computing" while in the edit mode to please the good folks in Camarillo, California.

dBASE allows you to have more than one index for a file. Suppose we wanted to have the data sorted by "name" *and* by ZIP code. This would be impossible using SORT, but it's a simple thing for INDEX. Type this:

> *"Another major advantage of INDEX over SORT is that you don't have to continually re-sort the file any time you add, delete, or edit a record."*

```
INDEX ON ZIP TO MAILZIP  ENTER
```

The "Mailname" index is still on the disk, but a new index has been created with the data sorted by ZIP code. List the names and ZIP codes to see for yourself:

```
LIST NAME,ZIP:CODE         ENTER
```

The list will be in order of ZIP code (Figure 4).

If we make changes to our database file now, the "Mailzip" index will be updated — but not the "Mailname" index. This problem is easily overcome by using two indexes at once. Only the first index you specify will be used for finding data and listing data, but the other index or indexes will be kept up to date any time changes, additions, or deletions are made. You can specify more than one index with the "SET INDEX TO" command if the file is already in use or when you open the file with the "USE" command:

```
SET INDEX TO MAILNAME,MAILZIP
```

This would cause the "name" index to be the main index. You can use FIND to locate names and commands such as LIST will output the data in name order. Any time a change is made, the ZIP code index is updated. You could also say:

```
USE MAIL INDE MAILNAME,MAILZIP
```

This works the same as the above example except it opens the file at the same time.

When you want the data in ZIP code order, just put the ZIP code index first:

```
USE MAIL INDE MAILZIP,MAILNAME
```

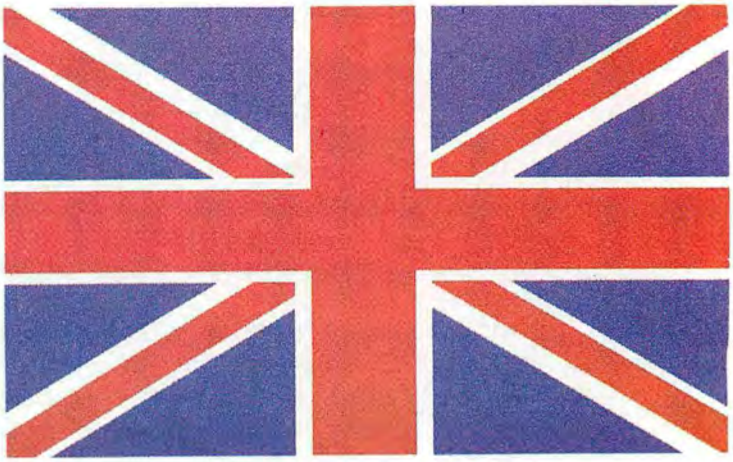
The ZIP code index is the main index now, and the name index just tags along.

As you can see, dBASE II packs in a lot of guts on mild mannered floppy disk. We've only just begun, my friends! Over the next several months, we'll discover even more of the power of dBASE II. Stay tuned . . .

The 2000 Gallery

Flag of the Month Club

By Wayne Sanders



Continuing our series showcasing the beautiful graphics capabilities of the Tandy 2000, this month we present the flag affectionately known as "Union Jack," the flag of the United Kingdom.

The British flag interestingly combines symbols of England, Scotland, and Ireland. The red cross of St. George was a national symbol of England since the 1200s. A white 'X' cross on a blue field, St. Andrew's cross, was a long time symbol for Scotland. St. Patrick's cross, a red 'X' on a white background, was associated with Ireland. Together, these three symbols create a truly original national flag.

This program reproduces the United Kingdon's flag on the color monitor of your Model 2000. If you have a CGP-220 color ink jet printer, you may use the *CGPDMP.BIN* utility from your MS-DOS disk to print the flag in full color.

To use *CGPDMP,* change the first line of the program to read:

1000 SP=1

If you're using *CGPDMP,* you'll need to reserve memory when you enter BASIC from MS-DOS. Use the following command line to get into BASIC:

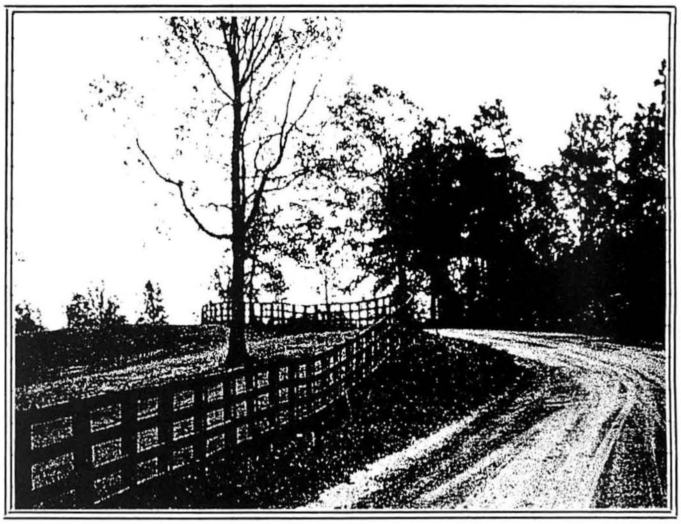
BASIC/M:&HFF00

The Flag of the Month Club is always interested in the fruits of your creative work. Choose a flag you find interesting or challenging to reproduce, perhaps your state flag, and write a program to draw it. Send it to use so we can share it with the world — we'll give you credit. Each time you do, you'll learn new little graphics programming techniques that will always be valuable.

The listing:

```
1000 SP=0
1010 IF SP THEN CGPDMP=&HFF00:BLOAD"CGPDMP.BIN",CGPDMP
1020 SCREEN 3:PALETTE 1,4:PALETTE 7,15:PALETTE 4,1
1030 CLS:KEY OFF
1040 LINE (0,0)-(639,399),4,BF
1050 LINE (0,0)-(0,40),7
1060 LINE (0,0)-(0,40),7:LINE -(599,399),7:LINE -(639,399),7
1070 LINE -(639,359),7:LINE -(40,0),7:LINE -(0,0),7:PAINT (1,1),7,7
1080 LINE (639,0)-(599,0),7:LINE -(0,359),7:LINE -(0,399),7:LINE -(40,399),7
1090 LINE -(639,40),7:LINE -(639,0),7:PAINT(638,1),7,7:PAINT (1,398),7,7
1100 LINE (0,0)-(0,30),1:LINE -(294,205),1:LINE -(320,190),1
1110 LINE -(5,0),1:PAINT (1,1),1,1
1120 LINE (639,399)-(639,369),1:LINE -(345,195),1:LINE -(320,210),1
1130 LINE -(635,399),1:PAINT (638,398),1,1
1140 LINE (639,0)-(614,0),1:LINE -(320,180),1:LINE -(340,200),1
1150 LINE -(639,20),1:PAINT (638,1),1,1
1160 LINE (0,399)-(30,399),1:LINE -(320,220),1:LINE -(300,200),1
1170 LINE -(0,380),1:LINE -(0,399),1:PAINT (1,398),1,1
1180 LINE (260,0)-(380,399),7,BF
1190 LINE (0,155)-(639,245),7,BF
1200 LINE (275,0)-(365,399),1,BF
1210 LINE (0,165)-(639,235),1,BF
1220 LINE (0,0)-(639,399),0,B
1230 IF SP THEN CALL CGPDMP(ER%)
1240 GOTO 1240
```

Logging Your Paycheck Can Be Helpful Down The Road



By Robert Frowenfeld
PCM Contributing Editor

It seems as if all I ever do is write programs for everyone. Everyone else but me, that is. I think I could safely say that in all my years of programming, I've written maybe five or six programs for myself. So you can imagine that those few would satisfy a very real need. In this month's "On the Road," I'll share with you one of these little "gems." It's a program to help you keep track of your paychecks. In doing so, it also monitors up to eight different payroll deduction categories.

One of the interesting features of this program is the fact that it will also work on your Model II, 12 or 16 computer. If you will notice on Line 2 of the accompanying listing, the variable MD is set equal to 100 (for the Model 100).

If you change this line to read: 2 MD=16 (etc.), you can get this program to run on your Model II, 12 or 16 without any modifications at all! If you decide to do this you'll find the results very attractive. Unlike many "modified" programs that just place the display in upper left corner of the Model II's screen, this one will put the display in the middle of the CRT, surrounded by an attractive block graphics border.

Now, for the working of *PAY.BA*, that's what I've called this handy little program. As you can see from the main menu, you can input or edit a record, as well as display or print category totals.

(Robert Frowenfeld owns his own computer programming firm in Louisville, Ky., and has completed his graduate course work in computer science at the University of Louisville.)

One quick note before you get started on this program. You must first use the *TEXT* program on your Model 100 to create a file named *PAY.DO* ; simply type in an asterisk (the character *) and press the F8 key to get back to the Model 100's main screen. The asterisk tells *PAY.BA* where the end of the data file is.

> *"One of the interesting features of this program is the fact that it will also work on your Model II, 12 or 16 computer."*

If you're running on a Model 2, 12 or 16 you will likewise have to create an output file. To do this, run this little program:

```
10 OPEN "O",1,"PAY/DAT"
20 PRINT #1,"*":CLOSE:END
```

The first option of *PAY.BA* lets you input your gross salary along with federal, state and social security deductions. You can see in Line 60 the data values for four other optional deduction categories. They have been set up for you as Ded #1 through Ded #4. If you wish to give these other names, just retype Line 60. You can even eliminate some or all of them by entering two quotes ("") for a field title. In this case, you will not be asked for input in these fields.

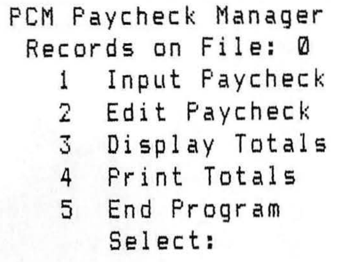
Option 2 lets you edit a record. All you have to do is enter the date of the check (be careful, 1/1/84 is not the same as 01/01/84) and it will be displayed on the screen for you to edit. The editing routine is very handy. First of all, the record number and net pay are displayed for you. The net pay feature is nice; it lets you verify that your gross salary minus deductions matches your net pay. When editing, just select the field number (0 to 9) and retype the value. I think you'll find this a very handy feature.

To give you your year-end totals, use option 3. Your year-to-date totals will be displayed almost instantly. This will certainly be helpful down the road, say around April 15! Oh yes, one more note. On Line 2, you will notice the variable NN is set equal to 24. That's because I get 24 paychecks each year. If you get paid weekly, you'll want to change this to 52, etc. Using 24 instead of 52 saves memory, because as you can see in Line 35, the DIM statement dimensions the various fields accordingly. Therefore, while having NN set to a higher value doesn't hurt anything, there's no need to use more memory than you have to!

Finally, the fourth option will direct a listing of each paycheck (along with deductions and totals for the end of the year) to your printer. I've set this up to fit into 80 columns, so you don't need a wide carriage to print it.

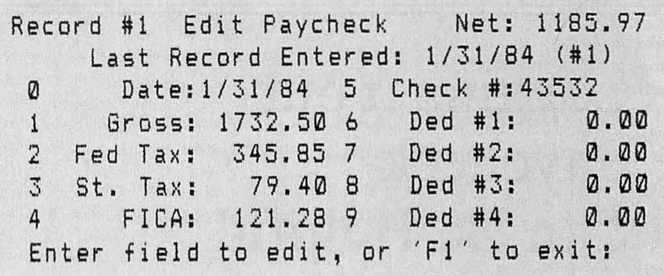
There you have it! Use it and enjoy. And may all your paychecks be deservingly large!

Figure 1
Main Menu



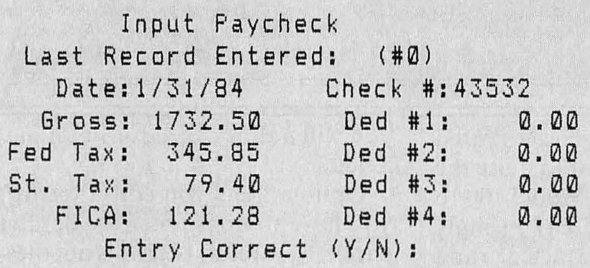
```
PCM Paycheck Manager
 Records on File: 0
    1  Input Paycheck
    2  Edit Paycheck
    3  Display Totals
    4  Print Totals
    5  End Program
       Select:
```

Figure 3
Editing an existing paycheck



```
Record #1  Edit Paycheck     Net: 1185.97
     Last Record Entered: 1/31/84 (#1)
 0      Date:1/31/84  5  Check #:43532
 1     Gross: 1732.50 6   Ded #1:     0.00
 2  Fed Tax:   345.85 7   Ded #2:     0.00
 3  St. Tax:    79.40 8   Ded #3:     0.00
 4      FICA:  121.28 9   Ded #4:     0.00
 Enter field to edit, or 'F1' to exit:
```

Figure 2
Adding a new record (paycheck)



```
         Input Paycheck
   Last Record Entered:  (#0)
     Date:1/31/84     Check #:43532
    Gross: 1732.50     Ded #1:     0.00
  Fed Tax:   345.85    Ded #2:     0.00
  St. Tax:    79.40    Ded #3:     0.00
     FICA:  121.28     Ded #4:     0.00
        Entry Correct (Y/N):
```

The listing:

```
1 CLEAR 500:CLS:DEFSTR A,R,U:DEFINT I-N
2 MD=100:NN=24
3 GOTO 15
4 FOR I=0 TO 7:IR=I:IC=0:A=BL$+" ":GOSUB
```

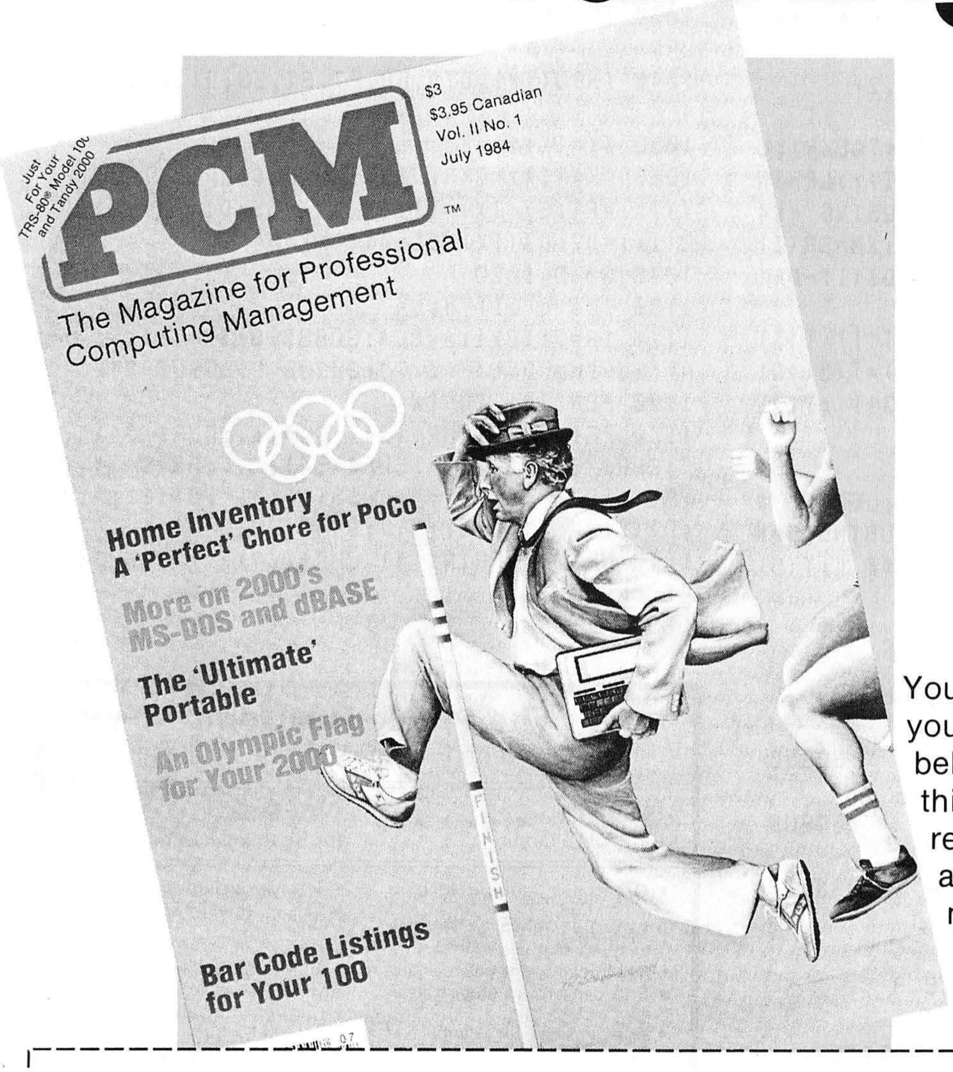
```
 8:NEXT I:RETURN 'clear screen
5 MD=16:GOTO 15
6 X=Ø:Y=Ø:IF ABS(FL)=1 THEN IN$=INPUT$(1
) ELSE LINE INPUT IN$
7 X=VAL(IN$):IF IN$<>"" THEN Y=ASC(IN$):
RETURN ELSE RETURN
8 IF MD=16 THEN PP=(IR+8)*8Ø+IC+2Ø ELSE
PP=IR*4Ø+IC
9 PRINT@PP,A;:RETURN
1Ø IF MD=16 THEN PP=(IR+8)*8Ø+IC+2Ø ELSE
 PP=IR*4Ø+IC
11 PRINT@PP,USING"#####.##";X;:RETURN
15 IF MD=16 THEN R=CHR$(26):U=CHR$(25) E
LSE IF MD=1ØØ THEN R=CHR$(27)+"p":U=CHR$
(27)+"q"
2Ø IF MD=16 THEN IR=-1:IC=-1:A=CHR$(128)
+STRING$(4Ø,15Ø)+CHR$(129):GOSUB 8:FOR I
=1 TO 8:IR=I-1:IC=-1:A=CHR$(148)+STRING$
(4Ø," ")+CHR$(148):GOSUB 8:NEXT I:IR=8:I
C=-1:A=CHR$(131)+STRING$(4Ø,15Ø)+CHR$(13
Ø):GOSUB 8
25 IF MD=16 THEN EE$="*" ELSE EE$="F1":K
EY 1,"*"+CHR$(13)
3Ø BL$=STRING$(39," ")
32 NN=24
35 DIM GR(NN),FT(NN),ST(NN),D1(NN),D2(NN
),D3(NN),D4(NN),DT$(NN),CK$(NN),SS(NN)
5Ø DATA "Input Paycheck","Edit Paycheck"
,"Display Totals","Print Totals","End Pr
ogram"
55 FOR I=1 TO 5:READ MØ$(I):NEXT I
6Ø DATA "  Gross","Fed Tax","St. Tax","
  FICA"," Ded #1"," Ded #2"," Ded #3","
Ded #4","Net"
65 FOR I=1 TO 9:READ DE$(I):NEXT I
9Ø GOSUB 1ØØØ
1ØØ GOSUB 4
11Ø IR=Ø:IC=9:A=R+" PCM Paycheck Manager
 "+U:GOSUB 8
12Ø FOR I=1 TO 5:IR=I+1:IC=12:A=R+STR$(I
)+" "+U+" "+MØ$(I):GOSUB 8:NEXT I
125 IR=1:IC=11:A="Records on File:"+STR$
(IX):GOSUB 8
13Ø IR=7:IC=16:A="Select: ":GOSUB 8
14Ø A=INPUT$(1):X=VAL(A):IF X<1 OR X>5 T
HEN 13Ø
15Ø FX=X:ON FX GOTO 2ØØ,3ØØ,4ØØ,5ØØ,6ØØ
2ØØ 'input
2Ø5 A1=R+" "+MØ$(FX)+" "+U:GOSUB 8ØØ
21Ø GOSUB 7ØØ:GOSUB 76Ø
212 DT$="":CK$="":GR=Ø:FT=Ø:ST=Ø:D1=Ø:D2
=Ø:D3=Ø:D4=Ø
215 GOSUB 23Ø:IF IN$="*" THEN GOSUB 11ØØ
:GOTO 1ØØ ELSE IR=7:IC=Ø:A=BL$:GOSUB 8:G
OSUB 235:GOSUB 24Ø:GOSUB 245:GOSUB 25Ø:G
OSUB 255:GOSUB 26Ø:GOSUB 265:GOSUB 27Ø:G
OSUB 275:GOTO 28Ø
23Ø IR=2:IC=12:A="":GOSUB 8:GOSUB 6
232 IF DT$="" AND IN$="" THEN 23Ø ELSE I
F IN$<>"" THEN DT$=IN$
233 IR=2:IC=12:A=STRING$(8," "):GOSUB 8:
A=DT$:GOSUB 8:RETURN
235 IR=2:IC=32:A="":GOSUB 8:GOSUB 6
237 IF CK$="" AND IN$="" THEN 235 ELSE I
F IN$<>"" THEN CK$=IN$
238 IR=2:IC=32:A=STRING$(7," "):GOSUB 8:
A=CK$:GOSUB 8:RETURN
24Ø IR=3:IC=12:A="":GOSUB 8:GOSUB 6
242 IF GR=Ø AND IN$="" THEN 24Ø ELSE IF
IN$<>"" THEN GR=X
243 IR=3:IC=12:X=GR:GOSUB 1Ø:RETURN
245 IR=4:IC=12:A="":GOSUB 8:GOSUB 6
247 IF FT=Ø AND IN$="" THEN 245 ELSE IF
IN$<>"" THEN FT=X
248 IR=4:IC=12:X=FT:GOSUB 1Ø:RETURN
25Ø IR=5:IC=12:A="":GOSUB 8:GOSUB 6
252 IF ST=Ø AND IN$="" THEN 25Ø ELSE IF
IN$<>"" THEN ST=X
253 IR=5:IC=12:X=ST:GOSUB 1Ø:RETURN
255 IR=6:IC=12:A="":GOSUB 8:GOSUB 6
257 IF SS=Ø AND IN$="" THEN 255 ELSE IF
IN$<>"" THEN SS=X
258 IR=6:IC=12:X=SS:GOSUB 1Ø:RETURN
26Ø IF DE$(5)="" THEN RETURN
261 IR=3:IC=32:A="":GOSUB 8:GOSUB 6
262 IF D1=Ø AND IN$="" THEN 26Ø ELSE IF
IN$<>"" THEN D1=X
263 IF DE$(5)="" THEN 26Ø ELSE IR=3:IC=3
2:X=D1:GOSUB 1Ø:RETURN
265 IF DE$(6)="" THEN RETURN
```

```
266 IR=4:IC=32:A="":GOSUB 8:GOSUB 6
267 IF D2=0 AND IN$="" THEN 260 ELSE IF
IN$<>"" THEN D2=X
268 IF DE$(6)="" THEN 265 ELSE IR=4:IC=3
2:X=D2:GOSUB 10:RETURN
270 IF DE$(7)="" THEN RETURN
271 IR=5:IC=32:A="":GOSUB 8:GOSUB 6
272 IF D3=0 AND IN$="" THEN 260 ELSE IF
IN$<>"" THEN D3=X
273 IF DE$(7)="" THEN 270 ELSE IR=5:IC=3
2:X=D3:GOSUB 10:RETURN
275 IF DE$(8)="" THEN RETURN
276 IR=6:IC=32:A="":GOSUB 8:GOSUB 6
277 IF D4=0 AND IN$="" THEN 260 ELSE IF
IN$<>"" THEN D4=X
278 IF DE$(8)="" THEN 275 ELSE IR=6:IC=3
2:X=D4:GOSUB 10:RETURN
280 IR=7:IC=10:A="Entry Correct (Y/N): "
:GOSUB 8:A=INPUT$(1):IF A="N" OR A="n" T
HEN 200 ELSE IF A$<>"Y" AND A<>"y" THEN
280 ELSE PRINT A;
285 IX=IX+1
290 CK$(IX)=CK$:DT$(IX)=DT$:GR(IX)=GR:FT
(IX)=FT:ST(IX)=ST:SS(IX)=SS:D1(IX)=D1:D2
(IX)=D2:D3(IX)=D3:D4(IX)=D4
295 GOTO 200
300 'edit
305 A1=R+" "+M0$(FX)+" "+U:GOSUB 800
```

```
310 GOSUB 700
315 FOR I=1 TO 10:IR=I+1:IF IR>6 THEN IR
=IR-5
320 IC=0:IF I>5 THEN IC=20
325 A=R+STR$(I-1)+" "+U:GOSUB 8:NEXT I
330 IR=7:IC=2:A="Enter Date of Check, or
 '"+EE$+"' to exit":GOSUB 8
335 IR=2:IC=12:A="":GOSUB 8
340 GOSUB 6:IF IN$="*" THEN GOSUB 1100:G
OTO 100 ELSE IR=7:IC=0:A=BL$:GOSUB 8
345 FOR I=1 TO IX:IF IN$=DT$(I) THEN 355
 ELSE NEXT I
350 IR=7:IC=3:A="Check not found, press
any key ... ":GOSUB 8:A=INPUT$(1):GOTO 3
00
355 DT$=DT$(I):CK$=CK$(I):GR=GR(I):FT=FT
(I):ST=ST(I):SS=SS(I):D1=D1(I):D2=D2(I):
D3=D3(I):D4=D4(I)
360 GOSUB 233:GOSUB 238:GOSUB 243:GOSUB
248:GOSUB 253:GOSUB 258:GOSUB 263:GOSUB
268:GOSUB 273:GOSUB 278
362 IR=0:IC=0:A="Record #"+MID$(STR$(I),
2):GOSUB 8:XN=GR-FT-ST-SS-D1-D2-D3-D4:IR
=0:IC=28:A="Net:":GOSUB 8:X=XN:IC=32:GOS
UB 10
365 IR=7:IC=1:A="Enter field to edit, or
 '"+EE$+"' to exit: ":GOSUB 8
370 IN$=INPUT$(1):IF IN$="*" THEN 385 EL
SE IF IN$<"0" OR IN$>"9" THEN 365 ELSE N
=VAL(IN$)
375 ON N+1 GOSUB 230,240,245,250,255,235
,260,265,270,275
380 GOTO 362
385 DT$(I)=DT$:CK$(I)=CK$:GR(I)=GR:FT(I)
=FT:ST(I)=ST:SS(I)=SS:D1(I)=D1:D2(I)=D2:
D3(I)=D3:D4(I)=D4
390 IF MD=100 THEN GOSUB 6
395 GOTO 300
400 'display totals
405 IF IX=0 THEN 100 ELSE A1=R+" "+M0$(F
X)+" "+U:GOSUB 800
410 GOSUB 700:IR=2:IC=1:A=BL$:GOSUB 8:IR
=7:IC=15:A="Computing ... ":GOSUB 8
415 FOR I=1 TO 8:T(I)=0:NEXT I
420 FOR I=1 TO IX
425 T(1)=T(1)+GR(I):T(2)=T(2)+FT(I):T(3)
=T(3)+ST(I):T(4)=T(4)+SS(I)
430 IF DE$(5)<>"" THEN T(5)=T(5)+D1(I)
435 IF DE$(6)<>"" THEN T(6)=T(6)+D2(I)
440 IF DE$(7)<>"" THEN T(7)=T(7)+D3(I)
445 IF DE$(8)<>"" THEN T(8)=T(8)+D4(I)
450 NEXT I
455 FOR I=1 TO 8:IR=I+2:IC=12:IF I>4 THE
N IR=IR-4:IC=32
460 IF I>4 AND DE$(I)="" THEN 470
465 X=T(I):GOSUB 10
470 NEXT I
475 GOSUB 770:GOTO 100
500 'print totals
```

```
505 IR=7:IC=0:A=BL$:GOSUB 8:IR=7:IC=14:A
="Printing ... ":GOSUB 8
510 LPRINT"  Date    ";:FOR I=1 TO 9:T(I)
=0:NEXT I
515 FOR I=1 TO 9:A=DE$(I)
520 IF LEFT$(A,1)=" " THEN A=MID$(A,2):G
OTO 520
525 J=LEN(A):IF J<8 THEN AA=STRING$((8-J
)/2," "):A=AA+A+AA:IF LEN(A)<8 THEN A=A+
" "
530 IF I=4 OR I=9 THEN A=MID$(A,2)
535 LPRINT A;:NEXT I:LPRINT
540 FOR I=1 TO IX:LPRINT DT$(I);TAB(9);U
SING "####.## ";GR(I);FT(I);ST(I);:LPRIN
T USING"###.## ";SS(I);:LPRINT USING"###
#.## ";D1(I);D2(I);D3(I);D4(I);:XN=GR(I)
-FT(I)-ST(I)-SS(I)-D1(I)-D2(I)-D3(I)-D4(
I):LPRINT USING"####.##";XN
545 T(1)=T(1)+GR(I):T(2)=T(2)+FT(I):T(3)
=T(3)+ST(I):T(4)=T(4)+SS(I):T(5)=T(5)+D1
(I):T(6)=T(6)+D2(I):T(7)=T(7)+D3(I):T(8)
=T(8)+D4(I):T(9)=T(9)+XN
550 NEXT I
555 LPRINT:LPRINT"Totals"TAB(8);USING" #
###.##";T(1);T(2);T(3);:LPRINT USING"###
#.## ";T(4);:LPRINT USING"####.## ";T(5)
;T(6);T(7);T(8);T(9)
560 GOTO 100
600 'end program
610 IF MD=16 THEN CLS:END ELSE MENU
700 'display screen
705 IR=2:IC=07:A="Date:           ":GOSUB
 8
710 IR=2:IC=24:A="Check #:        ":GOSUB
 8
715 FOR I=1 TO 8:IR=I+2:IF IR>6 THEN IR=
IR-4
720 IC=4:IF I>4 THEN IC=24
730 A=DE$(I)+":          ":GOSUB 8
740 NEXT I
750 IR=1:IC=5:A="Last Record Entered: "+
DT$(IX)+" (#"+MID$(STR$(IX),2)+")":GOSUB
 8
755 RETURN
760 'F1
765 IR=7:IC=7:A="Enter "+R+" "+EE$+" "+U
+" for DATE to exit":GOSUB 8:RETURN
770 'continue
775 IR=7:IC=5:A="Press any key to contin
ue ... ":GOSUB 8:A=INPUT$(1):RETURN
800 'clear screen @ print title
810 GOSUB 4
820 IR=0:IC=20-LEN(A1)/2
830 A=A1:GOSUB 8
840 RETURN
900 'open for input
910 IF MD=16 THEN OPEN"I",1,"PAY/DAT" EL
SE OPEN "PAY.DO" FOR INPUT AS 1
920 RETURN
930 'open for output
940 IF MD=16 THEN OPEN"O",1,"PAY/DAT" EL
SE OPEN "PAY.DO" FOR OUTPUT AS 1
950 RETURN
1000 'read in all data
1010 GOSUB 900:IX=0
1020 IF EOF(1) THEN CLOSE:RETURN ELSE IN
PUT #1,CK$
1030 IF CK$="*" OR EOF(1) THEN CLOSE:RET
URN
1040 INPUT #1,DT$,GR,FT,ST,SS,D1,D2,D3,D
4
1050 IX=IX+1
1060 CK$(IX)=CK$:DT$(IX)=DT$:GR(IX)=GR:F
T(IX)=FT:ST(IX)=ST:SS(IX)=SS:D1(IX)=D1:D
2(IX)=D2:D3(IX)=D3:D4(IX)=D4
1070 GOTO 1020
1100 'save all data
1110 IR=7:IC=1:A=BL$:GOSUB 8:IR=7:IC=12:
A="Saving Data ... ":GOSUB 8:GOSUB 930
1120 FOR I=1 TO IX
1130 PRINT #1,CHR$(34);CK$(I);CHR$(34)",
";CHR$(34);DT$(I);CHR$(34);",";GR(I);","
;FT(I);",";ST(I);",";SS(I);",";D1(I);","
;D2(I);",";D3(I);",";D4(I)
1140 NEXT I:PRINT #1,CHR$(34);"*";CHR$(3
4):CLOSE
1150 RETURN
```

Is your subscription running out?


$3
$3.95 Canadian
Vol. II No. 1
July 1984
Just For Your TRS-80® Model 10
and Tandy 2000
PCM
TM
The Magazine for Professional Computing Management
Home Inventory A 'Perfect' Chore for PoCo
More on 2000's MS-DOS and dBASE
The 'Ultimate' Portable
An Olympic Flag for Your 2000
Bar Code Listings for Your 100
FINISH

You'll still have time to catch it if you fill out and return the coupon below — or the card bound into this issue. Either way, you'll receive another full year of PCM at a savings of 22% off the newsstand price!

PCM — The Magazine for Professional Computing Management
9529 U.S. Highway 42
P.O. Box 385
Prospect, KY 40059

(502) 228-4492

□ Please renew my subscription to **PCM** □ NEW

Name ______________________________

Address ______________________________

City ______________ *State* ______________ *ZIP* __________

□ Payment Enclosed Charge □ VISA □ MasterCard □ American Express

Account Number ______________ Interbank # (MC only) ______________

Signature ______________ Card Expiration Date______________

Subscriptions to **PCM** are $28 a year in the United States. Canadian and Mexican rate is U.S. $35. Surface rate elsewhere U.S. $64. Air mail U.S. $85. All subscriptions begin with the current issue. Please allow 5-6 weeks for first copy.

SOFTWARE

Open Your 2000 To The Excitement Of *Open Access*

By Lawrence Falk

Integrated software comes to the Tandy 2000 with a vengeance through *Open Access*, one of the most complete software packages we have seen for this, or any other, sophisticated computer system. Indeed, *Open Access* is one of the best single buys you can make in making your 2000 into a full-blown business and management system.

The full package from Software Products International — SPI — includes modules for database management, a spreadsheet, word processing, graphics (including 3-D graphs), communications and time management. Each of these programs is a complete system in itself, but the primary feature of *Open Access* is that each is part of an integrated system which allows one part to transfer data and information to another.

As an example, suppose you create a database containing the names, addresses, start dates, social security numbers, salaries and departments of all employees in your company. Once the database is created, the information can be transported to a spreadsheet and further moved into graphics, all without ever entering the data again.

Need to incorporate the same information into a letter? That can be done with the word processing module. Want it transferred to a remote computer? The communications module can do the job.

At first blush, *Open Access* seems to be quite formidable. It contains two thick manuals — a tutorial and a reference guide — plus a "getting started" booklet and several other things. The original thought might well be "How will I ever learn all of this?", but, in truth, it is important to consider that what you are dealing with is six full-blown programs.

Consider that and the documentation isn't quite as frightening. In fact, it is a lot easier to learn the six *Open Access* programs than it would be to learn six separate programs. That is because most of the conventions which apply to one program apply to all of them — and many keys are common to all. In truth, this really "whittles down" *Open Access* quite quickly.

As something of an expert in looking at documentation (I review a lot of software and hardware), I would rate that of *Open Access* in the Top 10 as far as ease of understanding is concerned. There are a few gaps along the way, but once I understood the database manager, I found the spreadsheet much easier to comprehend. As a teacher might say, I had already learned something.

I won't try to go into total detail on any of the modules, because they all work quite well. The database manager is easy to learn and quite flexible. It makes reports extremely easy to produce. You can set up data entry screens as you wish, use a default report or customize your own, do mailing labels and the whole bit. It searches and sorts quickly and efficiently. *Open Access* database manager is a program well worth buying for itself.

Like other modules, the database manager uses windows to allow use of information from more than one database at a time. This can be very important in such applications as accounting, receivables and the like. This is an important option that makes *Open Access* perfect for a wide range of applications.

The spreadsheet is also easy to use. In fact, one of the easiest and fastest I have seen. Those who think that all spreadsheets are alike should take a look at *Open Access*, which lets you concentrate on the problem rather than how you enter the problem.

Having worked with a great number of spreadsheets from *VisiCalc* on up, I would rate *Open Access* one of the very best. This is just from the standpoint of number crunching, the very thing for which spreadsheets were designed. *Open Access* does an excellent job with the "what if's" that are the *raison d'etre* of all spreadsheets.

But there is more. Suppose what you really need to know is what you have to do to net $1 million during your next fiscal year. With other spreadsheets, the only way to do this is to keep playing with the numbers — changing this and that — until your forecast comes out as you want it. *Open Access*, however, provides a handy "goal seeking" method: You enter in what you want to achieve, and the program tells you what the "what if's" have to be.

Maybe we should not quibble. After all, anyone who ever sat down with a piece of accounting paper and a calculator to try to work out a complicated projection will rightly believe that a spreadsheet where you only have to change the variables and have the calculations done for you is the greatest thing since sliced bread. But, now, you don't even have to work out the variables — just tell *Open Access* what you want to accomplish and it will do the goal seeking for you. A truly marvelous feature!

I cannot say enough about the graphics. You have a choice of so many different kinds of graphs — bar, pie, line and 3-D — that you have to feel your cup runneth over. Additionally, you can change colors and textures of graphs all at once or — even — produce a "slide show" of one graph after another.

Can you print them out? Yes, you can and with a variety of printers that are common to most computer systems.

But the *piece de resistance* is the 3-D graphics. These are beautiful and allow all sorts of data to be clearly shown. In addition, the graphics are drawn quickly and, here's that word again, "easily." I constructed a 3-D graph to show subscription projections for PCM in 15 minutes, including the time it took me to enter the data. Yes, you can enter the data directly into the graphics module, for a "quick and dirty" visual representation of information.

The word processing module does everything a word processor should be able to do. And, again, this is accomplished with ease. By the time I got to the WP module, I was already familiar with many of the command keys common to all the programs, which made learning it very simple. And, the integration of word processing with the database manager means that you can produce form letters which are personalized with all sorts of variables. You can also import data from a spreadsheet and produce it in a letter or a report.

I also found the communications module easy to use. It has several advantages, such as storing commonly use telephone numbers, but it does the two things a communications program is supposed to do: Transfers data from one computer to another and allows you to access bulletin boards as well as subscription services like CompuServe, The Source and Dow Jones.

The time management option is the finest I have seen. It allows you to look at a whole month at a glance, set up days in 30 minute increments and even lets you write notes to yourself. And, if you have a tennis date every Monday at 6 p.m. for the next six months, you can place all of them into your schedule with one entry!

Open Access provides outstanding screen formats which are easy to use and read especially when combined with a color monitor. More than any other program I have seen thus far, it takes full advantage of the colors, speed and resolution available in the Tandy 2000.

Several of the features are in the "gee, I wish" category. A "calculator" is available in all the modules — which allows you to do simple addition, subtraction, multiplication and division on-screen. How many times have you been writing a letter when you had to stop to fetch a calculator just to figure what 18 percent of 9,457 was? You can do it on-screen with *Open Access* — and then zip it into the place you want in your letter.

The "Help" screens are truly helpful. You simply press F1 at any time you want some explanation. Most important, the information you get is really *helpful*. It is obvious that considerable time went into the preparation of this part of the program.

I am also impressed with the part of the documentation which explains how to set up your 2000 for *Open Access*. Considering how complicated the program is, this was one of the easiest installations I have been through. And, the one place where problems always seem to crop up — customizing to your equipment and tastes — came off with nary a hitch.

If it seems I am enthusiastic about *Open Access*, I am. *Open Access* is an outstanding example of integrated software which will make you — and your Tandy 2000 — into the kind of productive team you envisioned when you bought your computer in the first place. If you want to see the true computing power available in the 2000 — then watch it run *Open Access*!

(Radio Shack, cat. no. 90-0114, $595)

SOFTWARE

Traveling Appointment Manager: Professional Program With Bonus

By D.E. Mitchell

When you purchase *The Traveling Appointment Manager* from Traveling Software, Inc., you get an unexpected bonus. The tape actually contains two related but distinct programs — *The Traveling Appointment Manager* and *The Traveling Things-To-Do Manager*. Each of the programs helps you organize your time, but in different ways.

If you are like me, the first thing you will do when you receive this impressive-looking package is to load the programs. At the start of the manual, it tells you that the tape contains three programs; the core Appointment Manager program *(APPMGR)*, the Appointment Manager file backup and report printer module *(AM-B/P)*, and the Things-To-Do Manager program *(TTDMGR)*. (Actually, it has three copies of each, in that order.) Unless you have a 32K, empty machine without disk video BASIC in it, you will find yourself in the same dilemma I did — they won't all fit. *APPMGR* uses 10,198 bytes, *AM-B/P* 4,517 bytes, and *TTDMGR* 10,009 bytes, so you would need approximately 25K of free memory just to load them.

Only after reading the documentation through, did I realize that you aren't supposed to load them all at once. Only *APPMGR* is required for the Appointment Manager program to function, and *AM-B/P* doesn't have to be loaded until you are ready to check your data file and/or print a report. Even then, it can be swapped with *APPMGR* at that time and work alone. The Things-To-Do Manager is not quite so big, and the author has combined the file management and print routine as part of the main program. So the most space you absolutely need at any one time is slightly over 10K for the program and some space in addition for data files (more on them later).

Since there are two distinct programs, I will discuss the features of each separately at this point.

The Traveling Appointment Manager

When you run *APPMGR.BA* to start the program you get the Traveling Software logo for a brief moment and then a calendar is drawn for the current month. It is driven by the Model 100's internal clock, so the current month is displayed (assuming your clock is properly set). If you have already entered appointments for the month, days that have appointments will be highlighted. If you have already entered appointments for the current day, a message will alert you to that fact when you start the program. The date is displayed in Year: Month: Day format. Changing the year or the month will change the calendar display. Changing the day or just hitting ENTER will give you that day's appointments in summary form, or the opportunity to enter new ones if none have been previously entered. You can then choose to add, change, delete, reschedule or view appointments in detail. Since you enter a start time and estimated duration for each appointment, the program is able to detect overlaps and warn you of them.

In the interests of memory conservation, you have the option of setting a retention period for your data files. This is a handy feature, since the Model 100's memory definitely needs conservation, and different people will have differing retention requirements.

The *AM-B/P* program is also used for memory conservation. From its menu you may choose the file management system, which tells you how many appointments you have stored and how big the file is (each appointment seems to take roughly 60 bytes of storage).

You may then choose to backup the file if you want. This is a rather unusual procedure. The file, called *AMDAT.DO*, is invisible to your Model 100's menu. The backup feature makes it visible and assigns it a name based on the date you created it, e.g., AMO430.DO if created on 30 April. This file may then be saved, killed, and even edited using normal Model 100 techniques. This file, or one previously saved and recalled, can then be restored to the current, invisible file if desired.

The other side of the *AM-B/P* program allows you to print a very nicely formatted appointment schedule spanning a designated date period. This is an excellent feature to provide you with a hard-copy of the upcoming weeks schedule for reference, to be updated as you go along.

The Traveling Things-To-Do Manager

The operation of this program is similar to the Appointment Manager. However, since the file management, printer and core program are combined, you must select what you want to do when you first run the program. If you elect to run the core program, you are then met with the monthly calendar. From here, you proceed to the appropriate date. In this program, however, instead of extensive detail on each appointment, you rank-order a to-do list by date (either numerically or alphabetically). An ordered listing may then be reviewed on the screen or printed as a hard-copy, "Things to-do" list.

As previously described for the Appointment Manager, you have the option of setting the record retention period, checking the size of the current file, making the file visible for storage, editing, etc. (converts invisible *TDDATA.DO* to TD+date, e.g., *TDO430.DO)*, or of restoring a file to "current" status.

General Comments

The documentation is provided in two forms, the written version and an oral presentation by the "Traveling Professor" on the reverse side of the tape. The written version is extremely professional and thorough. Other than jumping the gun in loading (my fault), I was able to effectively use the program in a short period of time. The oral presentation was a bit too "cutesy" for my taste, and I really didn't feel that it contributed to the overall value of the program.

Since the programs were written before Radio Shack announced the disk/video interface, you can't really fault Traveling Software for the fact that it doesn't anticipate that upper memory will be different on the Model 100 when disk BASIC is resident. Disk BASIC resides in locations 58263 through 62960, so the CLEAR 256,62890 statement in Line 1 of both *APPMGR.BA* and *TTDMGR.BA* is illegal and results in a program abort with an "FC" error. Since the only way I know of to remove disk BASIC once it is booted is a complete power down, I needed to get around this somehow. Changing the statement to a simple CLEAR 256 (no location) in both programs seemed to do the trick, but I hesitate to recommend it since I'm not sure that somewhere down the line it might not unexpectedly bite you. Short of analyzing the entire program though, that was the only way I found to get it to work. Maybe the author can suggest a more scientific approach to get it to work with disk BASIC.

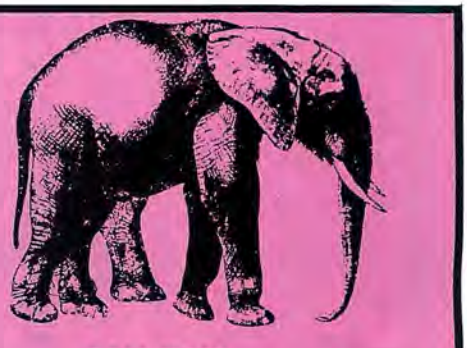
While on the subject of disk/video interface, the somewhat unusual method of file backup fits in quite nicely with it. Since you have the datafile under control of the Model 100 editor rather than having it stored and retrieved directly from a tape, you can easily store and retrieve it from a disk as well. I keep all three programs and my datafiles on a disk, and just load and carry around with me what I need at any given time.

One little quirk I noted but didn't know what to do about, was the retention of two small invisible files after the file backup procedure had been run, including a 115 byte *AMDATA.DO* and an 87 byte *TDDATA.DO* file. I could find them with a directory monitor utility, but they were invisible to the Master Menu and consequently to the "KILL" command. Not really a big thing, but curious.

Overall, I was very pleased with the total package. It did everything the authors said it would, and included the little additional "bells and whistles" that distinguish a truly professional program from one which is merely good. For example, when you have an appointment for the day, the program greets you on powerup with a Good Morning, Afternoon or Evening based on the clock's time of day. Not strictly necessary, but nice.

If you have a lot of meetings and appointments to schedule and keep track of during the day, then I feel that this software will definitely help you in keeping them straight. I have gone from not even knowing that I needed it to really missing having it when it isn't loaded. As with anything else, whether it is worth the asking price depends on your needs for its features. The quality of the programming is certainly consistent with its $59.95 price tag.

(Traveling Software, Inc., 11050 Fifth Avenue N.E., Seattle, WA 98125, 1-800-343-8080, $59.95)



BOOK

The Ideabook — Problem Solving With BASIC

You bought your Model 100 for portability, ease of use, and the built-in software you needed. You're not a programmer and never intended to be one. This computer spares you the pains and trivia of operating system commands and programming errors. But now you've mastered the applications software and feel confident enough to explore BASIC. You should consider David Ahl's *The TRS-80 Model 100 and NEC PC-8201 Ideabook* as your guide.

First, a word about what this book is not. It is not an exploration of Model 100 software other than BASIC. It never delves into the hardware wonders that made this machine *InfoWorld* magazine's "Hardware of the Year." It doesn't attempt to offer any insights into the unique abilities of your portable BASIC. Pretty basic, you say? Exactly! So what's the big *idea*?

The Ideabook introduces you to problem-solving concepts using Model 100 BASIC. Each problem is presented with a brief prose introduction, a program listing ready to type and run on your trusty computer, and sample output. But the real idea often comes last. The author challenges you to apply the techniques just demonstrated to solve a similar problem or explain and modify your results. You're probably already flipping to the back of the book for answers. But wait; it isn't going to be that easy. The idea's the thing *you* supply.

The 50 samples cover everything from a simulation of a lunar landing to a quadratic equation solver. The programs are divided into nine categories:

1) Drill and Practice
2) Problem Solving
3) Sets and Repetitive Trials
4) Convergence and Recursion
5) Compounding
6) Probability
7) Geometry and the Calculus
8) Science
9) Potpourri

All of the sample programs should run on an 8K Model 100 with memory to spare.

Most of the programs are short, averaging 10 to 20 lines each. A simple 11-line interest program demonstrates that if only you could invest $100 at 10 percent interest, you'd have $259.37 at the end of 10 years. In 12 lines of BASIC, you learn that a dollar can be changed 292 different ways. More importantly, you are asked to alter that solution for new circumstances. Suppose two of the coins are quarters. What if four of the coins are dimes? If you answer the questions, the programs become a real education, and an approach gradually emerges.

Besides the programs and the approaches, *The Ideabook* offers a wealth of interesting trivia already hinted at. Ever wonder about the term for "a word, verse, or number that reads the same backwords or forwards"? The section on palindromes provides the answer. It also provides a fascinating way to generate a palindrome from most numbers. And it challenges you to discover what numbers will not become palindromes using this method.

As the author of 16 books and founder of *Creative Computing* magazine, David Ahl brings considerable experience to *The Ideabook*. The explanations and descriptions are well written and very easy to follow. If you are new to Model 100 BASIC, that fact, and the fact that these programs run as listed on your PoCo make this a better choice than a general treatment of the BASIC language.

If you hope to unlock advanced functions like telecommunications and graphics, this volume does not hold the key. The only reason I can imagine for this omission is that the Model 100 version of *The Ideabook* is only one of a series of six. To take advantage of the BASIC similarities among such different machines as the Commodore 64, the Epson HX-20, the TI-99/4A, and Microsoft BASIC (of which Model 100 BASIC is one dialect), the author has chosen the least common denominator as his focal point. The same examples, with minor revisions, work for six different BASICs. Still, I would very much appreciate a section that covers sound, graphics, and communications in the context of practical problem solving. At the very least, these capabilities could be demonstrated beyond the two simple plot routines included.

The Ideabook is a worthwhile course in computer problem-solving. Once you've worked your way through the book with computer in lap, you know a lot more of what your Model 100 can and cannot do. You learn to make your machine do more and do it better. That is, after all, the whole idea.

(Creative Computing Press, 39 E. Hanover Avenue, Morris Plains, NJ 07950, Softbound, 139 pages, $8.95)

— Dennis Kirley

SOFTWARE

Radio Shack's *Scripsit 100:* Fast, Simple To Use

Radio Shack has finally released its version of a text-processing program for the Model 100. *Scripsit 100* takes text files created within the Model 100 *TEXT* program and reformats them according to your specifications.

The program is distributed on cassette. Although simple, the installation procedure (creating a backup copy of the tape and installing the RAM copies of the program files) is quite time consuming. The manual requires almost four pages to describe the process. Once installed, however, the program and associated modules can easily be run from MENU. The entire set of programs requires only about 4K of RAM storage.

The *WSPEC.DO* file which accompanies the program contains default formatting specifications which *Scripsit 100* uses when printing your text files. It provides initial settings for: lines per page (page length); line length; top, bottom, left and right margins; number of printed lines per page; header and footer information; line format (normal, justified or centered); spacing between lines; whether an extra line should be printed following each paragraph; first and last pages to be printed; first page number; number of copies; continuous form or single sheets; whether a form-feed should be issued at the end; where output should be directed; and a printer setup string. *WSPEC.DO* can be modified through the *TEXT* program as your word-processing style and default requirements change.

Let's examine a few of the *WSPEC*

.DO options in more detail. First, the header line, in addition to containing user-specified text strings, has provisions for automatic inclusion of the date, time and/or page number. Both headers and footers can easily be toggled on or off from within *WSPEC.DO* — without erasing the header/footer information. Second, you can print a subset of each document since both the starting and ending page numbers can be specified. Third, since you can indicate the first page number to be printed it is possible to join documents together — printing a second one where the first left off. Fourth, the "Output to:" item allows a formatted file to be printed to a printer, the screen, cassette, or to COM: (a serial printer or another computer). By printing to LCD:, for example, you can preview the document on screen and save paper.

Of course, if the *WSPEC.DO* defaults were to remain constant throughout the printing of the document, *Scripsit 100* would be fairly limited. To allow some control over basic formatting, the program offers 11 types of "dot" commands which can be embedded within your text. These allow you to change the line formatting (right justified, centered, or normal — ragged right); toggle headers and footers off or on; reset the right and left margins; issue page breaks; suppress the printing of orphan lines at the bottom of a page; skin 'n' lines; and change the spacing between lines (single-, double-, or triple-spaced). Each dot command (e.g., ".C" for centered text) must be separated from other text by carriage returns.

Printer control commands (for boldfaced print, underlining, etc.) are supported through the Model 100's "Ctrl-P" command. This, of course, requires you to know what control or escape sequences are needed for each printer command. Not particularly convenient, but at least the feature exists.

Of the few Model 100 text-processing programs that I have examined so far, I prefer *Scripsit 100*. It is fast (no noticeable pauses while text is being formatted), appears to be bug-free, and is simple to use. In addition to its well-written manual, two pocket guides are included which offer assistance in copying the tape and using the dot commands. Until a full-featured word-processing program for the Model 100 becomes available — and I advise against holding your breath while waiting —*Scripsit 100* is my choice.

— Steven Schwartz

SOFTWARE

MAI Accounts Receivable 'One of the Best'

By Mary Jeann Batham

Nothing is closer to an accountant's heart than a good accounts receivable software program. MAI has managed to produce a very user-friendly instruction manual as well as an easy-to-use accounts receivable system.

This powerful program will handle 75 customer accounts and 300 open invoices on a single floppy diskette, or 175 customers with 1,750 open invoices on a two-disk system.

The software also ages accounts receivable for up to 120 days, works on either a cash or an accrual basis, records dates the accounts were opened and date of last payment, and ensures a consistent debit/credit balance with automatic control during data entry.

Since the program interfaces with MAI's General Ledger, it posts sales and cash receipts to the proper ledger accounts. The oldest invoice for a customer can be posted, or a specific invoice paid can be identified. Both credits and debits for an invoice/customer are input, and one-time-only customers are accommodated.

Either balance forward or detailed invoices receivable can be used, aged open invoice reports or customer invoice detail are displayed, and after-the-fact invoice data can be entered.

Printers supported by this software are the Radio Shack models DMP-200, DMP-400, DMP-420, DMP-500 or DMP-2100. I tested the program using the DMP-120 and old, faithful Line Printer VI, and both worked fine. The manual states that patches for other printers are being developed and that a new disk will be sent when the work is completed.

At the present time, the printer paper must be manually reset to top-of-form. Software to eliminate the form feed problem will be sent later.

I was very impressed with these features of the program:

Numbers can be input with decimals. (Many accounting programs don't recognize a decimal point, and if one is typed, serious errors occur.)

Screen formatting is excellent. A broken line appears on the CRT, indicating the length of the field. If field length is exceeded, rather than truncating the information, the program returns the cursor to the beginning of the line.

At the end of data input, the prompt, "Is the data correct?" appears. This saves the effort of editing the information later since corrections can be made instantly while the data is on the screen.

The program will allow you to use either preprinted (addressed) statements, or it will print company name and address, allowing the use of more economical standardized invoice forms.

Late charges can be personalized for each customer, standardized for all customers, or ignored. The late charges option can be transported to the General Ledger, or added as a memo to customer statements and not posted to the Ledger.

The file sizing is good. Default file sizes can be accepted, or values can be input. The program advises if a second data disk is needed to accommodate the files.

Sales tax amounts can be set up in midyear and the program automatically adds tax figures to the original entries.

A customer cross-reference lists names by customer code on the same screen as an alphabetical listing. This helps weed out duplicate listings for a customer.

Displays can be output to the screen for a fast perusal of customer files, or a hard copy of the information can be printed.

Customer mailing labels are created from the program. Labels for all customers can be printed, or those for a selected few, defined by starting and ending costomer code, division, or customer name.

Whoever wrote the instruction manual actually knew how to use the program. Menus are easy to understand, and the main menu gives three choices: 1) set up the system (called Accounts Receivable definition), 2) post invoices and update files (Accounts Receivable system), and 3) maintenance (utilities).

The instruction manual follows the menus step-by-step and is divided into six chapters, three of which follow the main menu. The three others are: "Introducing the system," "starting up," and "Appendices," which contains accounting terms, error codes and hard disk information.

Flowcharts help you understand how the system works and Appendix A is an excellent tutorial which explains basic accounting concepts. The table of contents and accounting glossary are very helpful.

Ten of the function keys are preprogrammed for next, prior, first, last, date, list, delete, end, yes and no. A pack of labels is included with the program to illusrate the function keys. Use of these keys really speeds up the execution of of the program.

Several error-traps are built into the software. For example, if you try to delete a customer who owes for an invoice, the program prompts: "You may not delete active customers. CR to continue."

No software is perfect for everyone, and I encountered a few bugs. I tried to backup the operating disk using COPY A:*.* B:, but the tree directory didn't copy. Using DISKCOPY A: B:, as explained in Chapter 9, works fine.

When I tried to install a customer file, and attempted to save to disk, I kept getting "ERROR 16 line 4150 program AI1MAL Consult documentation." In the addendum, I was advised that error 16 was an error in CONFIG.SYS and was told to type END, <CR> and press Reset. That didn't solve the problem, and I kept looping back to the same situation.

Finally I went back to DOS, and typed:

```
COPY CON CONFIG.SYS
FILES = 10
^Z
```

This solved the problem, I was able to enter new customer files, and never encountered error code 16 again.

When correcting data, I tried to use the cursor keys, but kept getting graphic symbols, rather than moving the cursor right or left. The addendum advises "remove the reference to the Cursor keys in the Keyboard section." Ah, one of the reasons I like the Tandy 2000 better than my IBM-PC clone, is because the cursor keys are separate from the numeric keypad. Maybe when they improve the printer, and provide top-of-form software patches, they will restore control to my cursor keys!

On several sub-menus, I tried to use the END key, rather than typing "END" or using function key F8. The program ignored my END key (even with the num-lock off), so I got used to using the function key.

I was extremely impressed with the report forms generated by this program. The sales journal, invoice printing, open invoice data, cash receipts journal and sales tax reports were excellent.

I've used a lot of accounting programs, but the MAI *Accounts Receivable* software is one of the best.

(MAI *Accounts Receivable,* Tandy Corporation, 300 One Tandy Center, Ft. Worth, TX 76102, $495)

HARDWARE

The Axonix ThinPrint 80: New Portable Printer For PoCo

By Jim Hawk
PCM Contributing Editor

Third-party support for the Model 100 is taking off — three different portable printers have recently been announced for the 100. First out of the gate is the Axonix Corp. of Salt Lake City, with the ThinPrint 80 — a lightweight dot-matrix thermal printer. A second model from the same firm will soon offer letter-quality print using a Japanese mechanism, and the third will feature virtually the same unit, but be powered by AC only. We were privileged to get one of the first ThinPrint units to torture test, and the results point to the day when a portable printer will become a part of many Model 100 "field gear" kits.

Axonix is not coy about admitting it's using a mechanism made in the U.S. by the 3M company. In fact, the box the printer arrived in had nothing but a huge "3M" logo on it. Axonix engineers have taken the basic printer and added the necessary electronics to make it compatible with the Model 100, as well as batteries and a charging circuit to turn the normally AC-powered device into a portable. How does it work? Once I got it set up (more on that in a moment) I was amazed by the speed and quietness of the printer, and easy legibility of the finished product.

Using 8-inch wide thermal paper, the ThinPrint 80 turns out text that rivals many office bound DMPs. One time saving feature: the printer buffer is quite large and returns cursor control back to you in a very short time, letting you get on to other things while the printer finishes its job. The ThinPrint is also a tough little unit: the case is rugged enough to take the inevitable abuse of traveling, and the built-in NiCad batteries easily last a full day's work. The drawbacks are common to all thermal printers: somewhat fuzzy letters that are slightly darker on one side of the page than the other and paper that needs uncurling. Working with wire service printers that use roll paper, I've perfected the technique for holding the paper at both ends and keeping it taut while bending it against the curve on the edge of a desk — inconvenient at best. The fuzzy letters actually end up making the print look less like dot-matrix, since each dot tends to blur into the other and gets more readable once you're accustomed to it. The real price/performance breakthrough becomes obvious when you unhook the charger (a transformer-in-the-plug unit like the 100's) and take the printer out on the road. The Axonix is built around a metal chassis yet weighs only four pounds, so it's easy to carry around in a briefcase . . . with room to spare for the 100. Having the option to print out some work while away from the home-base printer came in handy —I finally got to type up newscasts for the radio station I work for, waiting till the last minute to "print out." And the unit's quiet, but impressive, performance leaves many onlookers amazed — especially when they ask why there's no line cord coming out of the unit.

The Model 100 and ThinPrint 80 are exactly the same width and nearly the same depth but the rear of the printer is about three-fourths of an inch higher in back, with a slanting front hood. Colors are the same light tan and black, with the tan printer cover extending to about two-thirds of the way back until it's met by the black plastic electronics/paper housing at the rear. The black part can be unlatched by releasing a pair of suitcase-like snaps on either side. It then folds back on hinges to allow changing of the special thermal paper rolls (at $4.20 a pop). This is also the place where I had to play electronics technician to get the Axonix printer working in the first place.

The 3M box in which the unit arrived contained the Axonix printer, a Centronics-type connector cable and a well-printed instruction book. Unfortunately, there were a couple of points left out of the instructions that had me phoning the company in Salt Lake City. Most important to the eager computerist: no one-page "quickie hookup" type instructions. Page 6 does mention the eight different interconnection cables offered for a variety of computers, but doesn't go on to tell how to physically connect any of them. In the case of the cable needed for the Model 100, it's not specified which way to plug the Centronics-style plug that can conceivably go in four different combinations from printer to computer. Most alarming of all to people who don't like to open up a brand new piece of electronics (much less make internal circuit changes) is that in order to get the Axonix to advance the paper you must make a circuit adjustment called for in Appendix A. However, the actual circuit board turned out to be significantly different from the one pictured. To figure out the correct procedure, I looked over the instructions and invoice sheet for either an 800 number or regular phone number. Surprisingly, the phone numbers were not to be found in the otherwise well-printed and documented instructions — a 50 cent call to AT&T's long distance operator yielded the information. All I had to do was unhook a certain jumper wire on the circuit board, but this should have been pre-set at the factory. Another thing that left me guessing: the Centronics connector that fits into the Model 100's printer port on the back has no indication of up or down — and the instructions made no mention about which way was correct. The parallel cable was an extra $40, on top of the $279 price for the printer itself and just a simple "up" sticker would have done the job. Not to be too critical with connectors never intended for the consumer market, I found it difficult to remove the plugs on either end of the cable.

For portable applications, a quick hookup is essential and it would be nice if some clever engineer figured out how to make such connections using a simple two-conductor audio cable — or maybe it's "impossible." Besides the text mode, there's also a graphics mode. But since the instructions were in computer-ese, and I'm no programmer, this feature went untested . . . example: "if the teleprinter protocol is selected, and the data byte is the ASCII 'Escape' character, then the next byte is examined to determine if the two bytes form one of the valid teleprinter protocol escape sequences." If you know what that means, then you're a prime candidate to translate the 32-page instructions. All in all, $249 is half what I paid for my AC-only dot-matrix, and the fact that the Axonix unit is battery operated makes it seem like a bargain.

One note: the company has already announced a more expensive portable printer utilizing a Japanese mechanism that creates letter-quality print through a unique five-wheel printer head. But it'll still be under $400 and use regular paper instead of thermal paper. A tractor-feed addition might be offered in the future, but for now it'll be page or continuous feed paper only. Axonix intends to keep selling their cheaper thermal printer as well, on the theory that not all people would want the slower speed (15 characters per second). And, another company called Dataport is already taking orders for a look-alike machine with a similar five-wheel printer head setup . . . it only operates off AC but the price is also about $100 less. More on these two letter-quality printers when they start being shipped.

(Axonix Corp., 417 Wakara Way, Salt Lake City, UT 84108, 801-582-9271, $279 includes batteries, AC adapter and an 80-page roll of paper)

New Products

PERCON Offers New Bar Code Reader

Peripheral Connections (PERCON) has introduced a newly enhanced bar code reader that will connect to any computer that supports RS-232 ASCII communications. The E-Z-READER bar code reader features the Hewlett Packard digital wand and can read the three most popular industrial codes: Code 3 of 9, Interleaved 2 of 5, and Codabar/ABC.

The E-Z-READER is designed to read dot-matrix printed bar codes, as well as those printed conventionally. The pen design enables it to read through thin protective plastic and poorly printed bar codes. Also, it can detect the code over a range of angles and pen speeds as well as variable bar code densities. A "good read" audible signal is a standard feature.

It is possible to use the E-Z-READER as a stand-alone terminal without a CRT. The host computer can be programmed to provide operator feedback. Two LEDs and a low frequency tone are provided for this purpose.

The E-Z-READER bar code reader is currently priced at $495. For more information, contact: Peripheral Connections, 2190 W. 11th St., Eugene, OR 97402; phone (503) 344-1189.

Electronic Mail

A new electronic mail software product is being offered for the Model 2000 by Omni Computer Systems. *FLASH-COM* comes with several modules including its own word/text processing module, forms/screen file management, mailing list manager, and communication module. It also contains a tutorial and dozens of standard business letters and forms ready to be used. It interfaces with applications written with *Lotus 1-2-3, dBASE II, PFS:File, Volkswriter, Perfect Writer, Perfect Calc, MultiMate, WordPlus,* and others, and it operates with more than a dozen modems currently available.

FLASH-COM is priced at $299 and interfaces with the U.S. Postal Service's E-COM service. For more information about the package, write to Omni Computer Systems Inc., P.O. Box 162, Chestnut Hill, MA 02167; or call 1-800-525-1055, (617) 277-2962 in Mass.

Quick Type Speeds Typing On 100

A program that speeds typing on the Model 100 and NEC PC-8201A has recently been introduced by PocketInfo Corporation of Beaverton, OR.

Quick Type lets users define abbreviations one to three characters long for commonly used words, phrases, sentences and paragraphs, providing a personal shorthand for users. The computer stores as many abbreviations as are needed, reducing the number of keystrokes necessary to type a document.

After typing a document with the abbreviations, the user runs *Quick Type* to expand the abbreviations into the correct full forms, producing a finished document for printing or transmission.

(continued on Page 54)

PCM BAR CODED LISTINGS

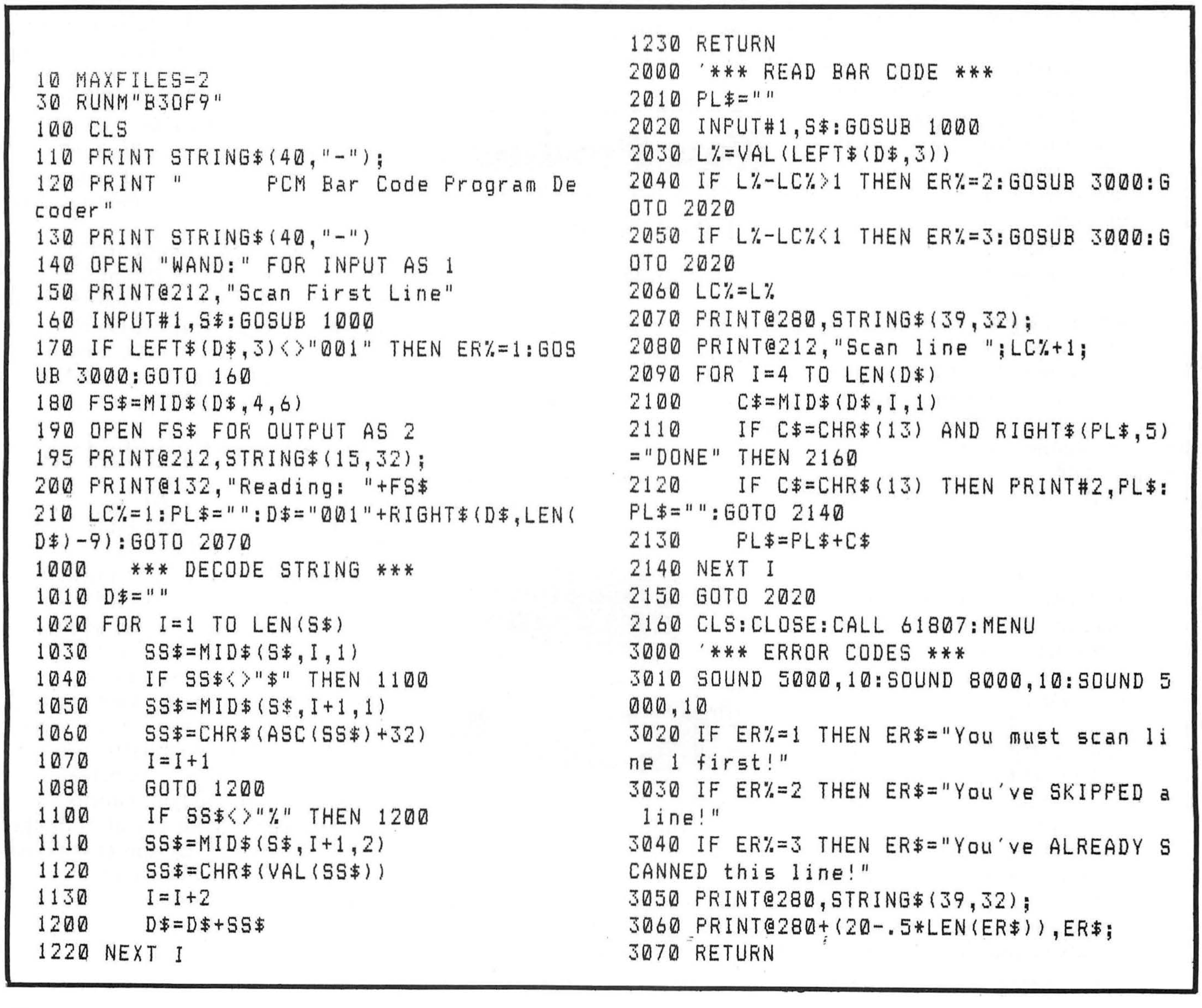
BASIC in Bar Code

Bar coded BASIC programs for the Model 100 can save hours of typing and correcting tedious program listings. Here's a refresher for those who are now familiar with how this works and/or do not have the necessary PCM Bar Code Decoder program.

You'll need a bar code wand and *B3OF9.CO* tape, the BASIC program listed here, and this copy of your favorite magazine.

Load *B3OF9.CO* from the cassette tape provided with your bar code wand following the instructions provided with the tape. Now, carefully type this program into your Model 100 and save it as "BAREAD."

It's really very simple to use. When it's RUN, the program will ask you to scan the first line of code (suggestions for successful bar code reading follow). The program name will be displayed on the screen and you will be asked to scan the second line. If you lose your place, a quick glance at the screen will get you back on track. *BAREAD* will make a "beep" sound after a successful scan and a "blip-bloop" sound if you scan the wrong line.



```
10 MAXFILES=2
30 RUNM"B3OF9"
100 CLS
110 PRINT STRING$(40,"-");
120 PRINT "       PCM Bar Code Program De
coder"
130 PRINT STRING$(40,"-")
140 OPEN "WAND:" FOR INPUT AS 1
150 PRINT@212,"Scan First Line"
160 INPUT#1,S$:GOSUB 1000
170 IF LEFT$(D$,3)<>"001" THEN ER%=1:GOS
UB 3000:GOTO 160
180 FS$=MID$(D$,4,6)
190 OPEN FS$ FOR OUTPUT AS 2
195 PRINT@212,STRING$(15,32);
200 PRINT@132,"Reading: "+FS$
210 LC%=1:PL$="":D$="001"+RIGHT$(D$,LEN(
D$)-9):GOTO 2070
1000    *** DECODE STRING ***
1010 D$=""
1020 FOR I=1 TO LEN(S$)
1030      SS$=MID$(S$,I,1)
1040      IF SS$<>"$" THEN 1100
1050      SS$=MID$(S$,I+1,1)
1060      SS$=CHR$(ASC(SS$)+32)
1070      I=I+1
1080      GOTO 1200
1100      IF SS$<>"%" THEN 1200
1110      SS$=MID$(S$,I+1,2)
1120      SS$=CHR$(VAL(SS$))
1130      I=I+2
1200      D$=D$+SS$
1220 NEXT I
1230 RETURN
2000 '*** READ BAR CODE ***
2010 PL$=""
2020 INPUT#1,S$:GOSUB 1000
2030 L%=VAL(LEFT$(D$,3))
2040 IF L%-LC%>1 THEN ER%=2:GOSUB 3000:G
OTO 2020
2050 IF L%-LC%<1 THEN ER%=3:GOSUB 3000:G
OTO 2020
2060 LC%=L%
2070 PRINT@280,STRING$(39,32);
2080 PRINT@212,"Scan line ";LC%+1;
2090 FOR I=4 TO LEN(D$)
2100      C$=MID$(D$,I,1)
2110      IF C$=CHR$(13) AND RIGHT$(PL$,5)
="DONE" THEN 2160
2120      IF C$=CHR$(13) THEN PRINT#2,PL$:
PL$="":GOTO 2140
2130      PL$=PL$+C$
2140 NEXT I
2150 GOTO 2020
2160 CLS:CLOSE:CALL 61807:MENU
3000 '*** ERROR CODES ***
3010 SOUND 5000,10:SOUND 8000,10:SOUND 5
000,10
3020 IF ER%=1 THEN ER$="You must scan li
ne 1 first!"
3030 IF ER%=2 THEN ER$="You've SKIPPED a
 line!"
3040 IF ER%=3 THEN ER$="You've ALREADY S
CANNED this line!"
3050 PRINT@280,STRING$(39,32);
3060 PRINT@280+(20-.5*LEN(ER$)),ER$;
3070 RETURN
```

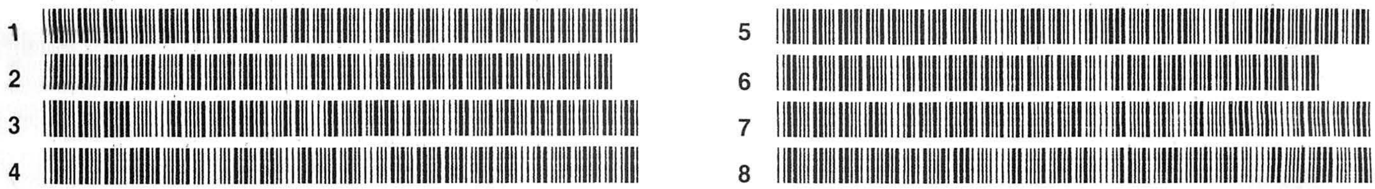
Before scanning any of the bar code listings contained in this section into your Model 100's memory, please be sure to carefully read the information on this page.

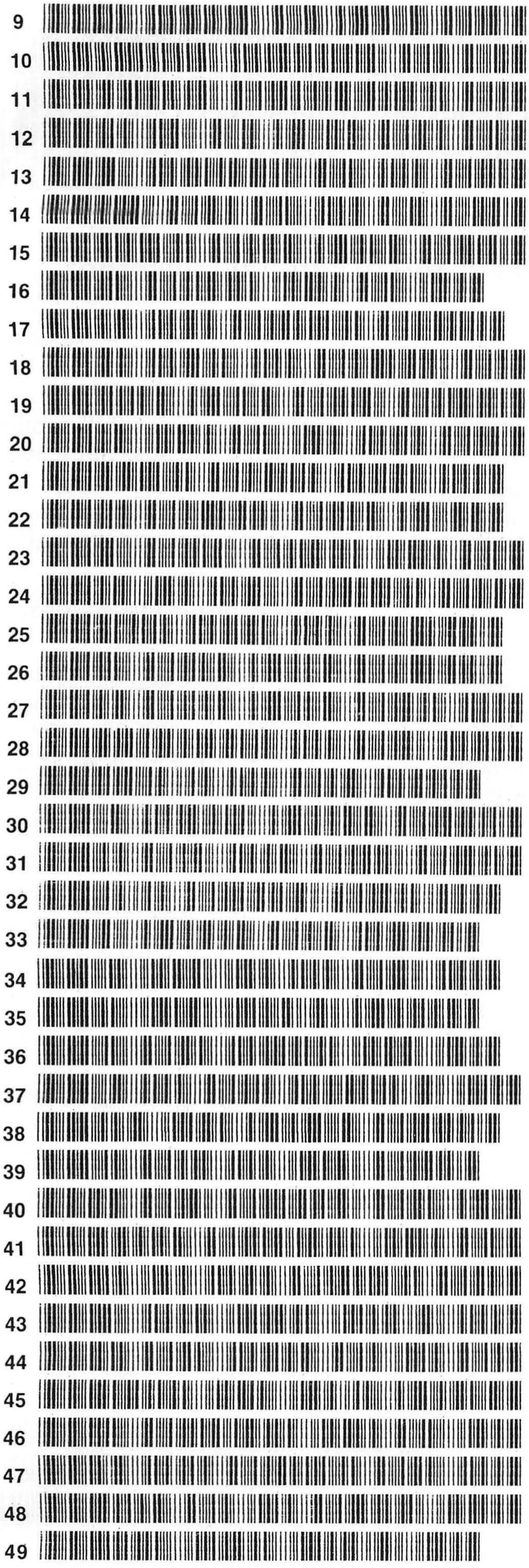
Tips for Successful Bar Code Reading

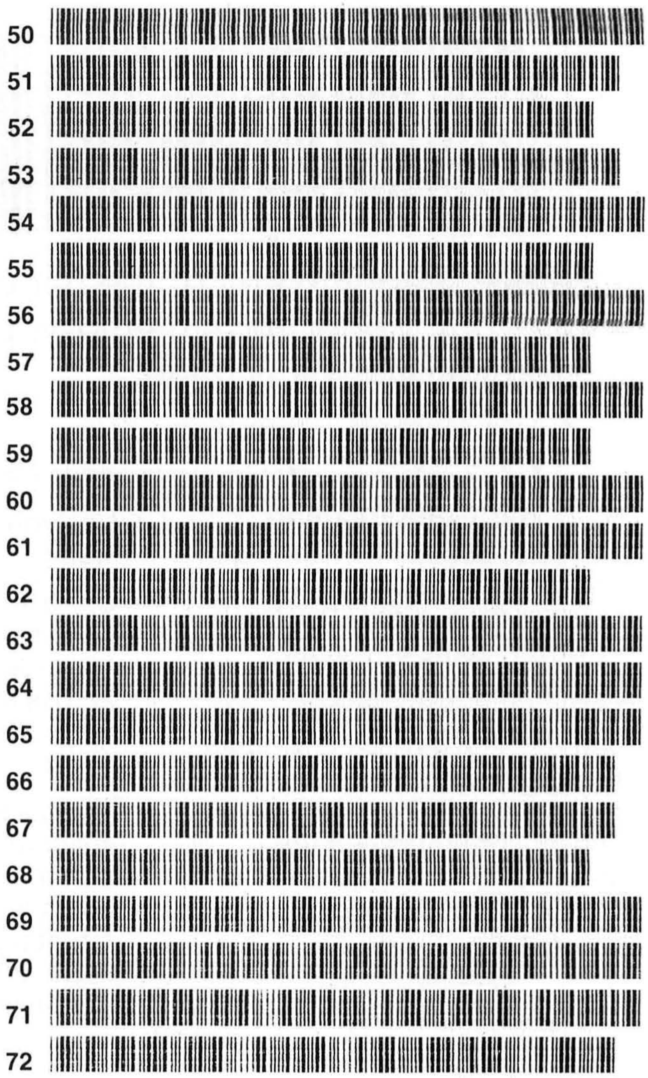
We are constantly working to improve the quality of the reproduction of PCM Bar Code; however, some of you may experience problems due to minor flaws in the bar code. As you become more accustomed to using the bar code wand, most of your problems will disappear. The following tips should help you become a successful bar code reader.

- Use an even, smooth speed.
- Keep a constant angle to the code (slightly less than 90°).
- Use your arm to move the wand — not your wrist.
- Use a straightedge such as a plastic ruler or an index card to guide your wand evenly across the code. A metal edge could damage your wand.
- If you experience problems scanning a line, move your straightedge slightly up or down and try reading a different part of the code — there may be flaws in the printing on the spot you're trying to read.
- Use a back-and-forth motion when a line does not read in on the first try.
- Make sure that the tip of your bar code reader is free of dust.
- Keep the LED on your bar code wand clean.
- Use a piece of non-reflective plastic, such as the one provided with your bar code wand, to protect the code from scratches.

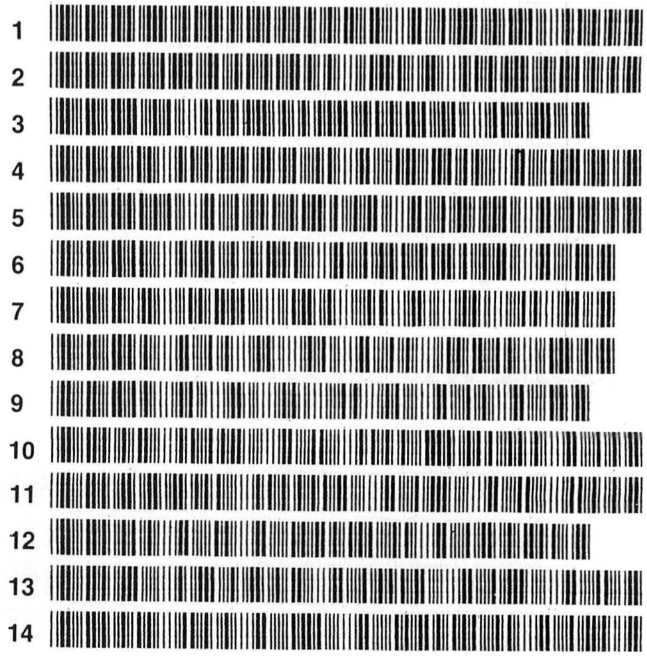
GRAF3D (FROM PAGE 30)

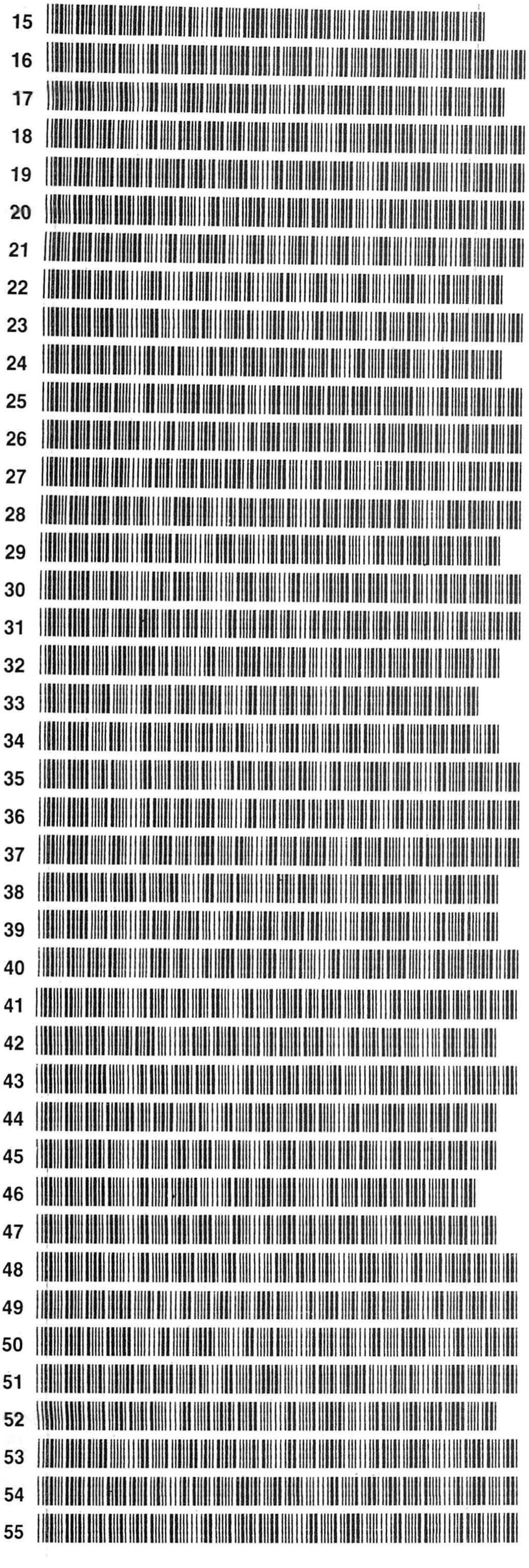


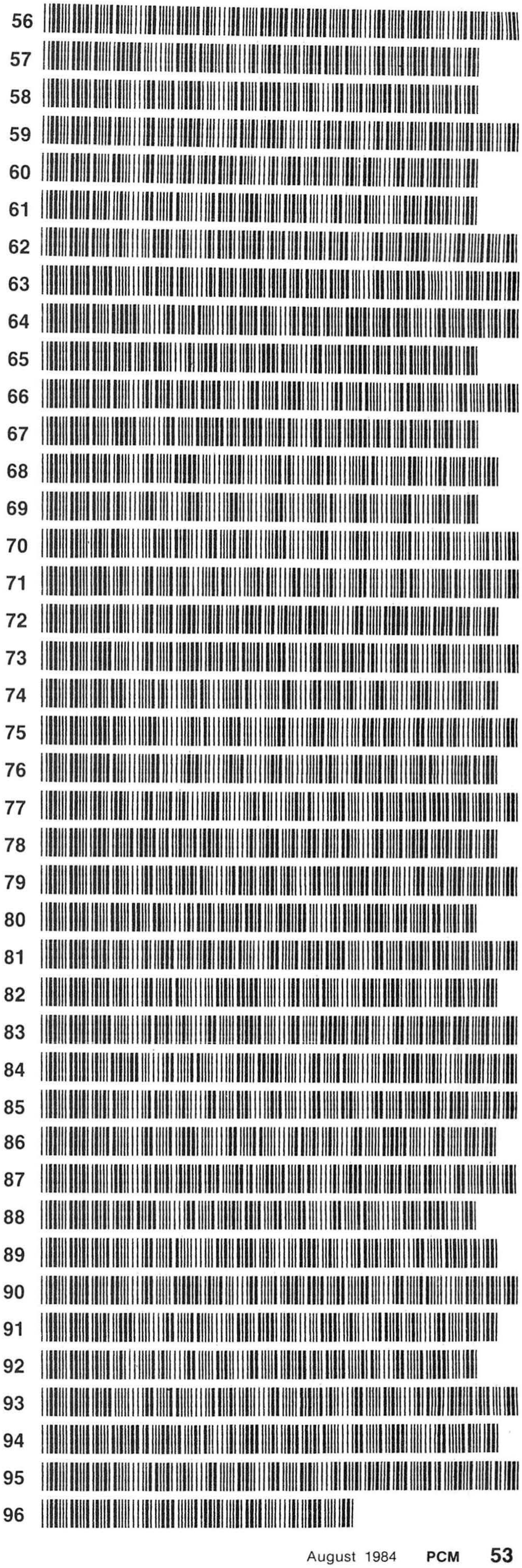




CIPHER (FROM PAGE 14)




15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55


56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96

The Rackseller

ALABAMA		
	Birmingham	Jefferson News Co .
	Florence	Anderson News Co.
	Madison	Madison Books
ALASKA		
	Fairbanks	Fairbanks News Agency
ARIZONA		
	Phoenix	Computer Pro
	Scottsdale	Softwareland Corp.
	Tucson	Anderson News Co.
CALIFORNIA		
	Citrus Heights	Software Plus
	Gretna	The Computer Supply Store
	Half Moon Bay	Strawflower Electronics
	Livermore	Software Galeria
	Lompoc	L & H Electronics Emporium
	North Hollywood	Levity Distributors
	San Francisco	News on 24
	Sunnyvale	Computer Literacy
DELAWARE		
	Wilmington	Normar Inc.—The Smoke Shop
FLORIDA		
	Boca Raton	Software, Software, Inc.
	Ft. Lauderdale	Software Connection
	Miami	Newsrack
	Panama City	Computer Systems Group
	Pensacola	Anderson News Co.
	Tallahassee	Anderson News Co.
GEORGIA		
	Atlanta	Guild News Agency
	Columbus	Muscogee News Co.
	Trenton	The Phone Line
ILLINOIS		
	Chicago	Kroch's & Brentano's Jackson Street
		Kroch's & Brentano's Wabash Ave.
	Chicago	Prairie News Agency
	Oakbrook	Kroch's & Brentano's
INDIANA		
	Elkhart	Elkhart City News & Book Store
	Indianapolis	A—Computer Store
	Mishawaka	Carrico's Radio Shack
	Scottsburg	Radio Shack of Scottsburg
IOWA		
	Ames	Myers TV & Stereo Repair
	Davenport	Interstate Book Store
KANSAS		
	Wichita	Amateur Radio Equipment Co.
		Lloyd's Radio
LOUISIANA		
	Slidell	Radio Shack
MAINE		
	South Portland	Portland News Co.
MARYLAND		
	Lexington Park	Books, Etc.
MASSACHUSETTS		
	Littleton	Computer Plus
	Marlboro	Radio Shack
MICHIGAN		
	Rochester	Rochester Book Center
	Sterling Heights	Software City
		Programs, Inc.
	Wyoming	Gerry's Book Co.
NEBRASKA		
	Lincoln	Hobby Town
NEW HAMPSHIRE		
	Manchester	Brookwrights
	Petersborough	WBR Electronics-Radio Shack
	West Lebanon	Verham News Corp.
NEW JERSEY		
	Hackensack	Total Circulation Service
	Marmora	Outpost Radio Shack
	Pennsville	Dave's Electronic Radio Shack
NEW MEXICO		
	Albuquerque	News and Bookstore
		Page One Newsstand
NORTH CAROLINA		
	Cary	It's Just For You, Inc.
	Raleigh	Softmart
NORTH DAKOTA		
	Fargo	Computer Associates
OHIO		
	Cincinnati	Cinsoft
	Toledo	Leo's
PENNSYLVANIA		
	Pleasant Hills	Pittsburgh Computer Store
RHODE ISLAND		
	Newport	Kelly's Variety
SOUTH CAROLINA		
	Greenville	Palmetto News Co.
	North Charleston	The Green Dragon
TENNESSEE		
	Knoxville	Anderson News Co.
	Memphis	Computer Center
	Nashville	Mosko's Book Store
	Smyrna	Delker Electronics, Inc.
TEXAS		
	Dallas	Micro Concepts, Inc.
	Ft. Worth	A & A International
		Byteworks, Inc.
		RFI Electronics
	Irving	Software Access
VIRGINIA		
	Norfolk	I-O Computers
WASHINGTON		
	Kennewick	Satellite TV Specialists
	Richland	C & J Electronics Computer Center
CANADA:		
ALBERTA		
	Calgary	Rainbow Software Services
	Edmonton	Kelly's Software Distributors
NOVA SCOTIA		
	Halifax	Atlantic News

Also available at selected B.Dalton Bookseller stores in the United States and Canada.

Users can create abbreviations while typing, and define and expand them later.

QuickType, which sells for $29.95, is available on cassette tape, and comes with a user manual. For further information, contact PocketInfo Corp., P.O. Box 152, Beaverton, OR 97075, (503) 649-8145.

Lotus 1-2-3 Available For The 2000

The *Lotus 1-2-3* integrated software system is now available for the Model 2000. It's carried by Radio Shack Computer Centers and participating dealers (Catalog Number 26-5300, suggested retail price $495).

Unlike versions of *1-2-3* for other computers, the Tandy 2000 program includes the MS-DOS operating system. There is no need to purchase MS-DOS separately at additional cost, and no "installation procedure" is required of the user.

The 2000's graphics allows *1-2-3* to use color in the text mode for highlighting in the "help" screens which the user may call up at any time there is a question regarding how to perform a function.

Lotus 1-2-3 includes a spreadsheet function capable of up to 2,048 rows of 256 columns, using all available memory. It features advanced cell and page formatting options, and statistical, financial, and calendar functions. Information from the spreadhseet can be used to create graphs, taking advantage of the Tandy 2000's high-resolution 640x400 monochrome and color graphics options. Bar and line graphs, pie charts, stacked bar charts, and X-Y graphs are available.

Graphs can be displayed on the screen in about two seconds, and may be printed using *PrintGraph*, a utility supplied with the program.

Lotus 1-2-3 requires at least 256K RAM, and may be used with the VM-1 High Resolution Monochrome Monitor (26-5111) with optional High Resolution Monochrome Graphics (26-5140), or the CM-1 Color Monitor (26-5112) with color graphics options.

New Portable Disk Drive For PoCo

Holmes Engineering Inc. of Murray, Utah, has released the Chipmunk™ — a 3½-inch portable disk drive for the Model 100. The Chipmunk comes ready to plug into your PoCo with no software installation required. Disk BASIC and a menu driven operating system are included. Power is provided by built-in rechargeable batteries or an AC adapter. Because of the small size of the Chipmunk, it can fit into a briefcase, along with the Model 100. Additional drives may be plugged into the Chipmunk to increase storage capacity. The Chipmunk with connecting cables, manuals and disk, retails for less than $550. Contact Holmes Engineering, 5175 Greenpine Drive, Murray, UT 84123, (801) 261-5652.

How To Put Ten Pounds Of ?✱÷▲!✿ In A Three Pound Package.

¡?✱÷▲!✱ is the stuff computer programs are made of. And now, Traveling Software has figured out a way to make your three pound lap-size computer do what only the big boys could do before.

Introducing T-base, the most powerful data base management system ever written for your TRS-80 Model 100, NEC PC-8201 or Olivetti M10.

T-base, a sort of place-for-everything-and-everything-in-its-place organizer, that allows you to keep multiple files of information or forms. But unlike simple filer programs, T-base is a relational data base manager. This allows you to borrow information that relates to one file from data contained in another. The net result is you can store more data by eliminating the unnecessary ?✱÷▲!✱ . And because T-base can link one file to another you have the ability to look at your data in a number of different ways. T-base also allows for the creation of custom printed or LCD display reports.

T-base has spreadsheet capabilities too! T-base allows you to build in your own formulas on your forms with automatic recalculations when any of your variables change.

T-base is easy. As with all Traveling Software programs, you not only get printed instruction, but you'll get an audio cassette with step-by-step prompting from the "Traveling Professor".

Simply put, T-base is the most important software ever written for your notebook computer. And because of its condensed format and broad data capabilities, your 3 pound package can do more than even you probably expect. T-base—it beats the ?✱÷▲!✱ out of the competition.

Special Introductory Price $79.95 — Save $20 through June 30.

TRAVELING SOFTWARE ™

Call now for the dealer nearest you.

Traveling Software, Inc.
11050 Fifth Avenue N.E.
Seattle, WA 98125

1-800-343-8080

Look for Traveling Software available for other lap-sized computers.

Traveling Software is now available in all Radio Shack stores via EXPRESS ORDER!

First Time Ever! Powerful Minicomputer Accounting Software Available on a Micro.

MAI Basic Four® Programs Ready-to-Run On the Incredible Tandy TRS-80® Model 2000

Our MAI/Basic Four accounting software is just what small businesses have been waiting for. Because it is quite simply the best accounting solution available today. Combined with our state-of-the-art Tandy 2000 Personal Computer, MAI/Basic Four programs will add power, flexibility and efficiency to every facet of your company's accounting procedures—general ledger, accounts payable, accounts receivable and much more. Since MAI/Basic Four software was originally designed for minicomputers, it's already been proven in thousands of businesses worldwide. All the programs are fully integrated, so you only have to enter data once. You can easily tailor the system to meet your special accounting needs. And the documentation is extremely clear and straightforward, so there's no previous computer experience required. Get a demonstration of the Tandy 2000 and MAI/Basic Four software today. Tandy 2000 systems start at $2999, or only $105 per month* on our commercial lease. Come see your accounting solution today!

Available at over 1200 Radio Shack Computer Centers and at participating Radio Shack stores and dealers.

Radio Shack® COMPUTER CENTERS
A DIVISION OF TANDY CORPORATION

*Plus applicable use/sales tax. Prices apply at Radio Shack Computer Centers and participating Radio Shack stores and dealers. MAI/Basic Four is a trademark of Management Assistance, Inc.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 1 PROSPECT, KY

POSTAGE WILL BE PAID BY ADDRESSEE

PCM

P.O. Box 385
Prospect, KY 40059